GEORGIA INSTITUTE OF TECHNOLOGY

OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

[x] ORIGINAL [ ] REVISION NO. ______

Project No. A-3786 GTRI/~~GIT~~ DATE 3/27/84

Project Director: J.L. Sims ~~School~~/Lab EML

Sponsor: U. S. Army Missile Command; Redstone Arsenal, AL 35898

Type Agreement: Delivery Order No. 0060 under Contract DAAH01-83-D-A013

Award Period: From 3/13/84 To ~~5/31/84~~ 9-29-84 (Performance) 7/31/84 (Reports)

Sponsor Amount: Total Estimated: $ 91,761.59 Funded: $ 91,761.59

Cost Sharing Amount: $ None Cost Sharing No:

Title: Submissile Aerodynamics Development, RDF-59

ADMINISTRATIVE DATA OCA Contact Linda H. Bowman ext. 4820

1) Sponsor Technical Contact:

Dr. M. M. Hallum
Systems Simulation & Dev. Directorate
US Army Missile Command
US Army Missile Laboratory
ATTN; ~~DRSMI~~ AMSMI-RDF
Redstone Arsenal, AL 35898

2) Sponsor Admin/Contractual Matters:

Mr. Thomas A. Bryant
Office of Naval Research
Room 206, O'Keefe Bldg.
Georgia Institute of Technology
Atlanta, Georgia 30332

Defense Priority Rating: DO-A2 under DMS Reg 1

Military Security Classification: Unclassified (general scope of work)
(or) Company/Industrial Proprietary:

RESTRICTIONS

See Attached Gov't Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval — Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category. See additional restrictions on re-budgeting travel funds: (2) C-2(f) (page 4 of basic agreement)

Equipment: Title vests with (1) B-2(c) (page 3 of basic agreement) Government; except that items costing less than $1,000 vest with GIT if prior approval to purchase is obtained from the Contracting Officer.

COMMENTS:

Send information copy of all correspondence addressed to Administrative Contracting Officer (ACO) to Commander, US Army Missile Command, Attn: DRSMI-ICLB, Redstone Arsenal, AL 35898

MAR 1984 RECEIVED Research Reports Office

COPIES TO:

Research Administrative Network	Research Security Services	Research Communications (2)
Research Property Management	Reports Coordinator (OCA)	Project
Accounting	GTRI	Other [illegible]
Procurement/EES Supply Services	Library	Other I. NEWTON

GEORGIA INSTITUTE OF TECHNOLOGY

OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION/CLOSEOUT SHEET

Date 12/10/84

Project No. A-3786

~~School~~/Lab EML

Includes Subproject No.(s) N/A

Project Director(s) J. L. Sims

GTRI / ~~GIT~~

Sponsor U.S. Army Missile Command; Redstone Arsenal, AL

Title Submissile Aerodynamics Development, RDF-59

Effective Completion Date: 9/29/84 (Performance) 11/29/84 (Reports)

Grant/Contract Closeout Actions Remaining:

- [ ] None
- [XX] Final Invoice or Final Fiscal Report
- [ ] Closing Documents
- [XX] Final Report of Inventions
- [XX] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other

Continues Project No.

Continued by Project No.

COPIES TO:

Project Director
Research Administrative Network
Research Property Management
Accounting
Procurement/EES Supply Services
Research Security Services
✓Reports Coordinator (OCA)
Legal Services

Library
GTRI
Research Communications (2)
Project File
Other A. Jones

A-3786

~~IN CONFIDENCE~~

Monthly Technical Report No. 1
and
Monthly Cost and Performance Report No. 1

Report Period
March 13 through April 30, 1984

Report Prepared
May 11, 1984

SUBMISSILE AERODYNAMICS DEVELOPMENT

Contract No. DAAH01-83-D-A013
Delivery Order No. 0060
GT/EES Project No. A-3786

Effective Date: 03/13/84
Expiration Date: 05/31/84

Prepared for

Commander
US Army Missile Command
Attn: DRSMI-ICD/Glass
Redstone Arsenal, AL 35898

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, GA 30332

WORK PERFORMED DURING THIS REPORTING PERIOD

The Subsonic Source Program has been activated and a number of cases for which experimental data is available in the SUBMIS Data Base have been run to develop guidelines for the placement of the sources/sinks to obtain the best solution. Also, modifications were made to the program to (1) calculate some input data which previously had to be calculated by the user, and (2) output the results of the calculations to a data file which can be read by the Subsonic Trajectory Program. The results of these modifications are to (1) minimize the user input data preparation time and effort and (2) provide an easily accessible means for transferring calculated data as input data into the trajectory program.

A review of the Subsonic Trajectory Program revealed there were unused elements which remained in the program after the wing and pylon calculations were removed. Program analyses are being conducted to develop plans for the deletion of all the unused elements to minimize program size. The program is operational and a few cases have been run with the source/sink data discussed above to demonstrate the operational capability of the trajectory program to calculate data for (1) physical trajectories, (2) Alpha sweeps, (3) X-sweeps, and (4) Z-sweeps. Plans are being formulated for inserting modifications to the program to read the data files created by the Subsonic Source Program. Analyses and plans for including fin lift curve slope calculations have been completed.

The Supersonic Trajectory Program is operational and has been exercised a number of times to investigate the effects of Dispenser Missile configuration on Submissile trajectories. This program also has unused and over-dimensioned elements left from the removal of the wing and pylon calculations. Plans and analyses are underway to define and remove all of these unused elements from the program.

PROBLEMS ENCOUNTERED

None.

WORK PLANNED FOR NEXT REPORTING PERIOD

Analysis of the Subsonic Trajectory program will continue and the unused elements will be deleted. Program modifications to read the subsonic source program output files will be completed. The subroutines necessary to calculate the fin lift curve slope will be integrated into the program and checked out.

The Supersonic Trajectory program analysis will be completed and unused program elements will be deleted. Analysis of the procedures and equations for calculating the fin lift curve slope at supersonic speeds will be initiated. Program modifications to simplify the geometric input data will be incorporated into the program and checked out.

Cost and Performance Report No. 1

Contract DAAH01-83-D-A013, D.O. 0060 GT/EES Project A-3786

The following charges have been incurred against the contract during the period March 13 through April 30, 1984.

	Expended	Encumbered
Personal Services (PS)	$13,456.63	$ -0-
Fringe Benefits	2,616.43	-0-
Materials and Supplies	8.45	-0-
Travel	-0-	-0-
Subtotal	$16,081.51	$ -0-
Equipment	-0-	-0-
Overhead (at 49.4% of Subtotal)	7,944.27	-0-
TOTAL	$24,025.78	$ -0-

The breakdown of personal services is as follows:

	Dollars	Approximate Manhours
Principal Research Scientists/Engineers	$ -0-	-0-
Senior Research Scientists/Engineers	8,072.43	318
Research Scientists II/Engineers II	2,338.25	116
Research Scientists I/Engineers I	-0-	-0-
Technicians/Draftsmen	-0-	-0-
Students	2,676.00	373
Secretarial/Clerical/Other	369.95	43
TOTAL	$13,456.63	850

The current financial status of the contract is as follows:

	Budget As Proposed	Expended	Encumbered	Free Balance
Personal Services	$51,090.78	$13,456.63	$ -0-	$37,634.15
Fringe Benefits	10,095.03	2,616.43	-0-	7,478.60
Materials and Supplies	234.27	8.45	-0-	225.82
Travel	-0-	-0-	-0-	-0-
Equipment	-0-	-0-	-0-	-0-
Overhead	30,341.51	7,944.27	-0-	22,397.24
FUNDING	$91,761.59	$24,025.78	$ -0-	$67,735.81

Based on present full funding, the funding and equivalent manhours are sufficient to complete the task. Approximately 73.8% of the proposed task has been completed.

Monthly Technical Report No. 2
and
Monthly Cost and Performance Report No. 2

Report Period
May 1 through May 31, 1984

Report Prepared
June 12, 1984

SUBMISSILE AERODYNAMICS DEVELOPMENT

J. L. Sims

Contract No. DAAH01-83-D-A013
Delivery Order No. 0060
GT/EES Project No. A-3786

Effective Date: 03/13/84
Expiration Date: 09/29/84

Prepared for

Commander
US Army Missile Command
Attn: DRSMI-ICD/Glass
Redstone Arsenal, AL 35898

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, GA 30332

WORK PERFORMED DURING THIS REPORTING PERIOD

Analysis of the Subsonic Trajectory program is in an advanced state. Most of the unused, unnecessary or redundant elements have been deleted to reduce the size of the program. The Subsonic Source program has been modified to write output files which contain the coefficients of the polynomials describing the body as well as the locations and strengths of the sources and sinks which represent the body. These files are read by the Subsonic Trajectory program. This reduces the volume of user required input data and makes the program much easier to use. Subroutines to calculate Submissile fin lift curve slope and center of pressure have been integrated into the program and have been checked out. With this addition and the transfer of geometric data from the source program, there are practically no calculations required to prepare input data from an engineering drawing.

Analysis of the Supersonic Trajectory program is also in an advanced state and it has been reduced considerably. The fin lift curve slope calculation procedure, which is much more complex than the subsonic case, has been programmed, checked out externally for all options, and then integrated into the trajectory program. The entire program was then exercised for one of the ASUBMIS configurations and it worked satisfactorily. A geometry subroutine has been added to the program to eliminate the calculations of the polynomial coefficients used to define the body segments.

A trip was made to the Martin Marietta Company in Orlando, Florida with Mr. Charles E. Brazzel, the Technical Monitor, to discuss the techniques and procedures of a Supersonic Trajectory program developed at MMC.

PROBLEMS ENCOUNTERED

None.

WORK PLANNED FOR NEXT REPORTING PERIOD

Both programs are now in an operational state and nearly optimized for the user. Sample calculations will be made for some of the cases in the experimental data base. Initial investigation of alternate methods for calculating the non-linear cross flow force will be begun. The analyses to minimize program operation time for special options and to further optimize the input schemes will continue.

Cost and Performance Report No. 2

Contract DAAH01-83-D-A013, D.O. 0060 GT/EES Project A-3786

The following charges have been incurred against the contract during the period May 1 through May 31, 1984.

	Expended	Encumbered
Personal Services (PS)	$ 8,076.13	$ -0-
Fringe Benefits	1,282.24	-0-
Materials and Supplies	-0-	31.50
Travel	-0-	483.00
Subtotal	$ 9,358.37	$ 514.50
Equipment	-0-	-0-
Overhead (at 49.4% of Subtotal)	4,623.03	-0-
TOTAL	$13,981.40	$ 514.50

The breakdown of personal services is as follows:

	Dollars	Approximate Manhours
Principal Research Scientists/Engineers	$ -0-	-0-
Senior Research Scientists/Engineers	4,934.25	194
Research Scientists II/Engineers II	-0-	-0-
Research Scientists I/Engineers I	-0-	-0-
Technicians/Draftsmen	-0-	-0-
Students	3,122.90	435
Secretarial/Clerical/Other	18.98	2
TOTAL	$ 8,076.13	631

The current financial status of the contract is as follows:

	Budget As Proposed	Expended	Encumbered	Free Balance
Personal Services	$50,483.78	$21,532.76	$ -0-	$28,951.02
Fringe Benefits	9,952.03	3,898.67	-0-	6,053.30
Materials and Supplies	234.27	8.45	31.50	194.32
Travel	750.00	-0-	483.00	267.00
Equipment	-0-	-0-	-0-	-0-
Overhead	30,341.51	12,567.30	-0-	17,774.21
FUNDING	$91,761.59	$38,007.18	$ 514.50	$53,239.91

Based on present full funding, the funding and equivalent manhours are sufficient to complete the task. Approximately 42% of the proposed task has been completed.

A-3786

~~IN CONFIDENCE~~

Monthly Technical Report No. 3
and
Monthly Cost and Performance Report No. 3

Report Period
June 1 through June 30, 1984

Report Prepared
July 23, 1984

SUBMISSILE AERODYNAMICS DEVELOPMENT

J. L. Sims

Contract No. DAAH01-83-D-A013
Delivery Order No. 0060
GT/EES Project No. A-3786

Effective Date: 03/13/84
Expiration Date: 09/30/84

Prepared for

Commander
US Army Missile Command
Attn: DRSMI-ICD/Glass
Redstone Arsenal, AL 35898

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, GA 30332

WORK PERFORMED DURING THIS REPORTING PERIOD

Optimization of the programs from the user viewpoint was continued as sample cases were run and the need for such changes became apparent. One of the major goals to be achieved from comparing sample case calculations with the experimental data is to develop a rational procedure to approximate the cavity flow field of open submissile bays in the dispenser missile with the source/sink distribution solution. Both the Subsonic and Supersonic Trajectory programs define the dispenser missile by dividing it into segments and using the conic section equation to define each segment shape. The original programs permitted a maximum of seven (7) segments to be used. This has always been sufficient to define any smooth body dispenser missile. However, it is not sufficient to define the complex flow configuration generated by a dispenser missile with three (3) open cavities such as the Data Base configuration. Program modifications are being made to increase the number of allowable segments to permit the definition of these complex shapes. These modifications to the Supersonic Trajectory program are nearly complete.

Sample Supersonic calculations which have been made, simulating the forward submissile bay only, show very encouraging results compared to the experimental data. These results raise the hope that a rational technique for approximating the cavity flow field can be developed.

PROBLEMS ENCOUNTERED

None.

WORK PLANNED FOR NEXT REPORTING PERIOD

Modification of the Subsonic Source and Subsonic Trajectory programs to use a larger number of segments to define the dispenser body shape will be completed. Sample case calculations will continue to be made with a two-fold purpose. First, it is desirable to establish the strengths and weaknesses of the programs for the calculation of submissile aerodynamic data with smooth body dispenser missiles. Second, the results of these calculations will be used to try to develop a rational technique for approximating the cavity flow. Investigation of alternate methods for calculating the non-linear cross flow force will continue.

Cost and Performance Report No. 3

Contract DAAH01-83-D-A013, D.O. 0060 GT/EES Project A-3786

The following charges have been incurred against the contract during the period June 1 through June 30, 1984.

	Expended	Encumbered
Personal Services (PS)	$ 8,458.32	$ -0-
Fringe Benefits	1,330.74	-0-
Materials and Supplies	84.73	-0-
Travel	491.75	-0-
Subtotal	$10,365.54	$ -0-
Equipment	-0-	-0-
Overhead (at 49.4% of Subtotal)	5,120.58	-0-
TOTAL	$15,486.12	$ -0-

The breakdown of personal services is as follows:

	Dollars	Approximate Manhours
Principal Research Scientists/Engineers	$ -0-	-0-
Senior Research Scientists/Engineers	3,799.37	150
Research Scientists II/Engineers II	-0-	-0-
Research Scientists I/Engineers I	1,057.60	64
Technicians/Draftsmen	-0-	-0-
Students	2,260.50	315
Secretarial/Clerical/Other	1,340.85	156
TOTAL	$ 8,458.32	685

The current financial status of the contract is as follows:

	Budget As Proposed	Expended	Encumbered	Free Balance
Personal Services	$50,483.78	$29,991.08	$ -0-	$20,492.70
Fringe Benefits	9,952.03	5,229.41	-0-	4,722.62
Materials and Supplies	234.27	93.18	-0-	141.09
Travel	750.00	491.75	-0-	258.25
Equipment	-0-	-0-	-0-	-0-
Overhead	30,341.51	17,687.88	-0-	12,653.63
FUNDING	$91,761.59	$53,493.30	$ -0-	$38,268.29

Based on present full funding, the funding and equivalent manhours are sufficient to complete the task. Approximately 58% of the proposed task has been completed.

A-3786

~~IN CONFIDENCE~~

Monthly Technical Report No. 4
and
Monthly Cost and Performance Report No. 4

Report Period
July 1 through July 31, 1984

Report Prepared
August 27, 1984

SUBMISSILE AERODYNAMICS DEVELOPMENT

J. L. Sims

Contract No. DAAH01-83-D-A013
Delivery Order No. 0060
GT/EES Project No. A-3786

Effective Date: 03/13/84
Expiration Date: 09/30/84

Prepared for

Commander
US Army Missile Command
Attn: AMSMI-ICDB/Glass
Redstone Arsenal, AL 35898

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, GA 30332

WORK PERFORMED DURING THIS REPORTING PERIOD

Modifications to the Subsonic Source and Trajectory programs, as well as the Supersonic Trajectory program, to increase the allowable number of dispenser missile body segments from seven (7) to fifteen (15) have been completed. This allows the open submissile bay cavities to be simulated with at least three (3) segments for each cavity and this should be sufficient for modeling the effective cavity flow bounding streamline. Early attempts to model the front cavity with very elementary streamline shapes have resulted in good qualitative agreement between available experimental data and calculated results. This approach, even though it is an indirect one, promises to yield submissile data which, at a minimum, have correct qualitative trends and, hopefully, will develop generalized approaches for modeling cavity flow streamlines. Some study of alternate methods for calculating the non linear cross flow forces has been done. The work to date has not shown any of the methods to be superior to the others. Therefore, no program changes have been made in this area.

PROBLEMS ENCOUNTERED

None.

WORK PLANNED FOR NEXT REPORTING PERIOD

Final computer program review will be completed. All desired program changes resulting from this review will be made. Initial planning for the final technical report will be completed. Further investigation of cavity modeling techniques will be conducted and the resultant calculations will be compared with experimental data. The investigation of alternative methods for calculating the non linear cross flow force will be concluded. Alternative methods will be incorporated into the programs if it is deemed advisable.

Cost and Performance Report No. 4

Contract DAAHO1-83-D-A013, D.O. 0060 GT/EES Project A-3786

The following charges have been incurred against the contract during the period July 1 through July 31, 1984.

	Expended	Encumbered
Personal Services (PS)	$ 6,128.57	$ -0-
Fringe Benefits	1,489.25	-0-
Materials and Supplies	-0-	-0-
Travel	-0-	-0-
Subtotal	$ 7,617.82	$ -0-
Equipment	-0-	-0-
Overhead (at 55.3% of Subtotal)	4,212.65	-0-
TOTAL	$11,830.47	$ -0-

The breakdown of personal services is as follows:

	Dollars	Approximate Manhours
Principal Research Scientists/Engineers	$ -0-	-0-
Senior Research Scientists/Engineers	3,509.41	113
Research Scientists II/Engineers II	-0-	-0-
Research Scientists I/Engineers I	2,538.22	152
Technicians/Draftsmen	-0-	-0-
Students	-0-	-0-
Secretarial/Clerical/Other	80.94	9
TOTAL	$6,128.57	274

The current financial status of the contract is as follows:

	Budget As Proposed	Expended	Encumbered	Free Balance
Personal Services	$50,483.78	$36,119.65	$ -0-	$14,364.13
Fringe Benefits	9,952.03	6,718.66	-0-	3,233.37
Materials and Supplies	234.27	93.18	-0-	141.09
Travel	750.00	491.75	-0-	258.25
Equipment	-0-	-0-	-0-	-0-
Overhead	30,341.51	21,900.53	-0-	8,440.98
FUNDING	$91,761.59	$65,323.77	$ -0-	$26,437.82

Based on present full funding, the funding and equivalent manhours are sufficient to complete the task. Approximately 71% of the proposed task has been completed.

A-3786

Monthly Technical Report No. 5
and
Monthly Cost and Performance Report No. 5

Report Period
August 1 through August 31, 1984

Report Prepared
September 25, 1984

SUBMISSILE AERODYNAMICS DEVELOPMENT

J. L. Sims

Contract No. DAAH01-83-D-A013
Delivery Order No. 0060
GT/EES Project No. A-3786

Effective Date: 03/13/84
Expiration Date: 09/30/84

Prepared for

Commander
US Army Missile Command
Attn: AMSMI-ICDB/Glass
Redstone Arsenal, AL 35898

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, GA 30332

WORK PERFORMED DURING THIS REPORTING PERIOD

A final review of the Subsonic Source program, Subsonic Trajectory program, and Supersonic Trajectory program has been completed. All technical program modifications have been completed, but further review to possibly simplify multiple case input data for second and subsequent cases is under way. Initial planning for the final technical report has been completed and the assembly of some of the report sections has been initiated. A reasonably successful cavity streamline model has been developed for one set of data. The non-linear cross flow force investigation has been completed. The alternative methods are not compatible with the submissile force calculation techniques and, therefore, will not be incorporated into the programs.

PROBLEMS ENCOUNTERED

None.

WORK PLANNED FOR NEXT REPORTING PERIOD

The final technical report will be written. Further calculations and comparison with experimental data will be made.

Cost and Performance Report No. 5

Contract DAAH01-83-D-A013, D.O. 0060 GT/EES Project A-3786

The following charges have been incurred against the contract during the period August 1 through August 31, 1984.

	Expended	Encumbered
Personal Services (PS)	$ 4,838.04	$ -0-
Fringe Benefits	1,175.64	-0-
Materials and Supplies	19.69	-0-
Travel	488.65	-0-
Subtotal	$ 6,522.02	$ -0-
Equipment	-0-	-0-
Overhead (at 55.3% of Subtotal)	3,606.68	-0-
TOTAL	$10,128.70	$ -0-

The breakdown of personal services is as follows:

	Dollars	Approximate Manhours
Principal Research Scientists/Engineers	$ -0-	-0-
Senior Research Scientists/Engineers	2,070.45	76
Research Scientists II/Engineers II	-0-	-0-
Research Scientists I/Engineers I	2,743.98	161
Technicians/Draftsmen	-0-	-0-
Students	-0-	-0-
Secretarial/Clerical/Other	23.61	1
TOTAL	$4,838.04	238

The current financial status of the contract is as follows:

	Budget As Proposed	Expended	Encumbered	Free Balance
Personal Services	$50,483.78	$40,957.69	$ -0-	$ 9,526.09
Fringe Benefits	9,952.03	7,894.30	-0-	2,057.73
Materials and Supplies	234.27	112.87	-0-	121.40
Travel	750.00	980.40	-0-	(230.40)
Equipment	-0-	-0-	-0-	-0-
Overhead	30,341.51	25,507.21	-0-	4,834.30
FUNDING	$91,761.59	$75,452.47	$ -0-	$16,309.12

Based on present full funding, the funding and equivalent manhours are sufficient to complete the task. Approximately 82% of the proposed task has been completed.

FINAL TECHNICAL REPORT

SUBMISSILE AERODYNAMICS DEVELOPMENT

By

J. L. Sims

Prepared for

U. S. ARMY MISSILE COMMAND
REDSTONE ARSENAL, ALABAMA 35898
ATTN: AMSMI- ICDB/Glass

Under

Contract No. DAAH01 - 83 - D - A013
Delivery Order No. 0060
Georgia Tech Research Institute Project No. A– 3786

November 1984

GEORGIA INSTITUTE OF TECHNOLOGY
A Unit of the University System of Georgia
Atlanta, Georgia 30332

GEORGIA TECH RESEARCH INSTITUTE

1984

Final Technical Report

SUBMISSILE AERODYNAMICS DEVELOPMENT

J. L. Sims

November 1984

Report Period: 13 March - 29 September 1984

Contract No. DAAH01-83-D-A013
Delivery Order No. 0060
Georgia Tech Project No. A-3786

Prepared for

U.S. Army Missile Command
Redstone Arsenal, Alabama 35898
AMSMI-ICDB/Glass

Prepared by

Georgia Institute of Technology
Georgia Tech Research Institute
Atlanta, Georgia 30332

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Submissile Aerodynamics Development	5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 13 MAR - 29 SEP 1984
	6. PERFORMING ORG. REPORT NUMBER A-3786-001
7. AUTHOR(s) J. L. Sims	8. CONTRACT OR GRANT NUMBER(s) DAAH01-83-D-A013 Delivery Order 0060
9. PERFORMING ORGANIZATION NAME AND ADDRESS Georgia Institute of Technology Georgia Tech Research Institute Atlanta, Georgia 30332	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Missile Command AMSMI-ICDB/Glass Redstone Arsenal, AL 35898	12. REPORT DATE November 1984
	13. NUMBER OF PAGES 218
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) U.S. Army Missile Command AMSMI-RDF/Hallum Redstone Arsenal, AL 35898	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Each transmittal of this document outside the agencies of the U. S. Government must have prior approval of the U.S. Army Missile Command, Redstone Arsenal, Alabama, AMSMI-ICDB/Glass.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Subsonic Flow, Supersonic Flow, Interference Aerodynamics, Multi-body Aerodynamics, Linear Theory, Slender Body Theory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report was prepared by the Electromagnetics Laboratory of the Georgia Tech Research Institute, Georgia Institute of Technology under contract DAAH01-83-D-A013. The U.S. Army Missile Command has a strong interest in calculating submissile trajectories and/or aerodynamic data for a single variable sweep when the submissile is separating from, or is in the presence of, a dispenser missile which may be flying at supersonic or subsonic speeds. Aircraft-store separation analyses and computer programs developed by Nielsen Engineering and Research, Inc., AFFDL-TR-76-46, Volumes I and II and

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

AFFDL-TR-72-83, Volumes I and II have been modified to specifically adapt them to these tasks. These computer programs were partially adapted to the sub-missile tasks by Lockheed Missile and Space Command LMSC-HREC TR 698338 which included the single variable sweep capability and adaptation of the programs to the Perkin-Elmer 8/32 computer. The original Nielsen programs contained very large elements to simulate the aircraft wings and the pylons which were used to support the stores. These elements are never needed for dispenser missiles in which the Army is interested and they substantially increase the size of the programs. The tasks which were performed and that resulted in the current set of programs that are documented herein are as follows: (1) The Subsonic and Supersonic Trajectory programs were to be reduced to minimum size by deleting all elements not required for dispenser missile simulation, (2) input data calculations were to be automated to the maximum possible extent to simplify input preparation, (3) revise the input format and overall plan to make the programs "user friendly" on the Perkin-Elmer 3230, and (4) modify the programs to create binary output data files that have the same format as the 'SUBMIS' data base files. Both programs have had all unused elements removed and the capability of performing either trajectory or single variable sweep calculations is available. Preparatory calculations of the fuselage body segment geometric coefficients, submissile fin lift curve slope, and submissile cross flow drag coefficients have been automated within the programs. All of the required input data now can either be obtained from an engineering drawing or selected to fulfill the user's requirement for type of calculation and output. The input data has been changed from a single formatted file to a combination of a Namelist file and a set of interactive input data. Division into the two types of input data was carefully selected to minimize the time and effort required to make sets of parametric calculations such as the data sets of the 'SUBMIS' data base. The Subsonic Trajectory program was configured for automatic transfer of source/sink distribution and geometry information from the required precursor Subsonic Source program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (cont.)

LIST OF TABLES

PREFACE

This report was prepared by the Georgia Tech Research Institute's Huntsville Operation, Georgia Institute of Technology, under Contract No. DAAH01-83-D-A013, Delivery Order No. 0060, for the U.S. Army Missile Command. The contract technical monitor was Dr. M. M. Hallum and the contract project monitor was C. E. Brazzel, both from the Systems Simulation and Development Directorate of the U.S. Army Missile Command. The contract period extended from 13 March through 29 September, 1984.

The views and conclusions in this document are those of the author and should not be interpreted as necessarily representing the official policies of the U.S. Army Missile Command.

1.0 INTRODUCTION

The U.S. Army Missile Command (MICOM) has a strong interest in large dispenser missiles which can eject small submissiles. One of the challenging technical problems of such a system is the calculation of the aerodynamic characteristics of a small body when it is within the region of the interference flow field created by the large body. This problem falls within the charter of the Aerodynamics Technology Branch of the Systems Simulation Directorate at MICOM. The Aerodynamics Technology Branch has developed a large experimental aerodynamic data base of forces and moments acting upon various submissiles in the presence of dispenser missiles at both subsonic and supersonic speeds. However, this branch also needed analytical tools for predicting the aerodynamic forces and moments acting upon a submissile in the presence of a dispenser missile and it also needed a simulation program to predict the trajectories of submissiles traversing the interference flow field of a dispenser missile. The aerodynamic programs would be used to predict submissile forces and moments which could be used to (1) compare with available experimental data to assess overall capability, (2) expand an available data base, and (3) create data bases for new configurations. The simulation program would either be used to predict the submissile separation characteristics as it traverses the dispenser missile flow field or, by using an inverse procedure, modify dispenser and/or submissile shape until the desired separation criteria are achieved.

Earlier work sponsored by MICOM in the development of these analytical tools are reported in References 1 and 2. Reference 1, which was the effort prior to the current one, discussed the methods, techniques, and solution elements which were contained in the original documented solutions of References 3 through 6. Also, the modifications introduced by Reference 2 were summarized and all of the modifications performed for Reference 1 were reported in detail. Therefore, only a terse summary of the program status at the beginning of this effort will be given herein.

The current effort started with programs that were operating and that were functionally correct. However, the programs contained many unused statements and elements. Many items which could have been calculated required input data with attendant, tedious hand calculations. While there are specific tasks called out in the Scope of Work which will be discussed in subsequent sections, two of the major goals of the present work are to make the programs (1) easier to operate and (2) more efficient to operate. The first goal has been met by (1) including calculation procedures for typical input items such as the polynomial coefficients describing each body segment, and by (2) arranging the input into a combination of Namelist data and interactive data input via the terminal keyboard to maximize utilization efficiency. However, the list-directed, formatted input file of the original program has been retained as an optional input capability which can be exercised if it is desired by the user. Significant effort was expended to (1) eliminate unused coding, (2) to optimize program operation from a computer science viewpoint, and (3) to assure that program operations not required by a selected operational mode are bypassed. It is felt that the programs which are documented in this report fully meet these goals.

Three separate computer programs are discussed and documented in this report. They are (1) the Subsonic Source distribution program, (2) the Subsonic Trajectory program, and (3) the Supersonic Trajectory program. Briefly, the three programs may be described as follows: The Subsonic Source program solves a set of simultaneous equations to determine the strengths of sources and sinks which represent the volume effects of the fuselages. This program must be run for each fuselage shape involved in problems for the Subsonic Trajectory program. The Subsonic Trajectory program will compute either a physical trajectory of a submissile separating from a dispenser missile or it will compute a single variable sweep of the submissile with respect to the dispenser missile (e.g., vertical separation distance) at a specified increment of the selected variable. This is the manner in which MICOM's "SUBMIS" data base was obtained (see Reference 7), and therefore this operational mode allows direct comparison of experimental and analytical data. The Subsonic Trajectory program computes five submissile component aerodynamic forces and moments at each trajectory integration time step or variable increment. The Supersonic Trajectory program operates in a manner similar to the Subsonic Trajectory program. However, it determines the source/sink representation of the dispenser missile volume effects internally and no precursor program is required.

The analysis and detailed computer program descriptions to follow will be arranged into sections for each of the three programs. Certain modifications were made in an identical manner to more than one of the programs. In these cases the modification is presented in detail with the first program under discussion to which it applies. Application of the modification to the other programs is indicated and then discussed under the other program.

2.0 SUBSONIC SOURCE PROGRAM

2.1 Program Description

The Subsonic Source program obtains the solution to a set of simultaneous equations to quantify the strength of the sources and sinks by representing the fuselage volume as a series of three dimensional point sources placed on the body axis. Once the source distribution has been determined, the program calculates the shape of the body represented by it. This body shape and the input body shape are printed at a selected number of points and if the source representation is not sufficiently accurate a new source distribution should be tried. It is important to note that if the fuselage is blunt based it is mandatory that the subsonic wake be approximated by a closed aft segment(s). The simulated wake must close to a sharp tip for the program to obtain a reasonable solution. However, for the calculations made to date, the solutions upstream of the simulated wake were not very dependent upon the assumed wake shape. Based on these results, a tangent ogive with an ℓ/d of 1.5 to 2.0 seems to be a usable approximation for the base wake. Two required user supplied input items which have a significant effect upon the body shape represented by the source distribution are the location of the first source along the body axis and the source spacing along the body length. An item of lesser importance to the source distribution is the location of the last source on the body axis. Calculations made to date have resulted in the development of some guidelines for these quantities which are discussed below.

2.2 First Source Location

There are two basically different requirements depending upon the nose type, i.e., either a sharp nose tip such as an ogive or cone, or a blunt nose such as a hemisphere. For the sharp nose body, the first source must be placed very close to the tip to retain the sharp tip feature and to keep the body derived from the source distribution from being larger than the real body. However, for the blunt nose body, the first source must be some distance from the tip to achieve the blunt feature. The version of the program documented in Reference 6 required the first source location to be input in terms of x/ℓ while the current program requires the input as x in whatever dimensional system is being used. Reference 6 gave some very rough guidelines for the first source location in terms of x/ℓ. Since various length afterbodies on a given nose shape should affect the location of the first source only in a secondary manner, it was decided to attempt a correlation of current calculations as a function of the nose length. Sharp nose solutions have been obtained for 1.5, 2.0, and 3.0 caliber long ogive noses on different length cylinders and the best solutions were obtained with the first sources placed at x/ℓ_N = 0.00207, 0.00180, and 0.00190. Therefore, for ogive nose bodies first source location x/ℓ_N = 0.00190 is recommended as an initial trial. This recommendation is for a tangent ogive-cylinder and the length of the ogive back to the cylinder is defined as ℓ_N. The main effect of the location of the first source is upon the radius of the body calculated from the source distribution at points near the tip of the body. A general result which has emerged from all solutions to date gives a qualitative guideline for moving the first source location to improve the solution near the nose tip. This guideline is: If the body radius calculated from the source distribution is larger than the body radius calculated from the polynomial equation, the first source should be moved forward. Conversely, if the body radius calculated from the source distribution is smaller than the body radius calculated from the polynomial equation, the first source should be moved aft. There are no precise guidelines for sizing the amount to move the first source location. If it is moved too far, the relationship between the source body radius and the polynomial body radius will reverse. However, quite often, small changes in the first source location will produce no change in the body radius computed from the source distribution. The reason for this apparent non-linearity has not been discerned from the solutions obtained to date.

Blunt nose (hemispherical cap) solutions have been obtained for the ASUBMIS submissile and the three BSUBMIS submissile configurations. For these four configurations, two solutions were found with first source locations, x/ℓ_N = 0.0390, and two solutions were found with first source locations, $x/\ell_N \approx$ 0.0750 and 0.0950. Review of the four solutions did not reveal specifically a reason for these differences. However, the overall solution for the blunt bodies must be affected by the fineness ratio of the cylinder and the assumption for some shape to simulate the wake. These solutions were obtained with different source spacing criteria on the hemispherical noses and this may be the major reason for the difference in the first source location.

All of the results and guidelines discussed above are based on the quality of the match between the source derived radius and the polynomial radius at the first body point location where these two values are calculated. A brief investigation was conducted to ascertain the effect of the longitudinal location of the first point that is used to make this judgement. This investigation was performed using a predetermined source distribution for each configuration while varying the first comparison point location from run to run. For blunt bodies (hemispherical nose cap), as long as the first body point is at about 12% or less of the hemispherical radius, the quality of the match between the two radii is essentially independent of the location of the point where the comparison is made. For the sharp nosed configurations, this effect was investigated using the Dispenser Missile configuration. When the comparison point location was in the order of 1.5 to 3% of the ogive length, the quality of the agreement between the two radii did not vary significantly. However, when the comparison point was moved forward to only 0.8% of the ogive length behind the tip, the tip of the sharp nose was not well represented. It is felt that this does not represent a physical problem as far as the source/sink strength effect on the flow field is concerned.

2.3 Source Spacing Criteria

The other user discretionary input data which can significantly affect the quality of the body fit is the source spacing criteria, PERCR(K). Once again, two sets of data seem to have developed; one for blunt nose bodies and one for sharp tip nose bodies. For the sharp nose bodies, a value of 0.7 was used for the nose ogive and a value of 1.0 for the cylinder and tail ogive. This gave a good overall fit between the source body and the polynomial body. These values might be made larger to reduce the number of sources and still obtain a good body match. Solutions for the hemispherical nose submissiles were obtained with PERCR(1) = 0.3 or 0.4 for the nose and a value of 0.8 for the cylinder and the simulated wake. Even this criteria resulted in only 6 to 8 sources along the length of the hemisphere; however, this was sufficient for good reproduction of the hemispherical shape. Guidelines for this criteria are soft; however, if this factor is made too small, resulting in the sources being very close together, the body shape derived from the source distribution oscillates badly.

2.4 Last Source Location

Experience to date has not shown the location of the last source to have any significant effect on the overall body coordinates derived from the source distribution. All of the calculations have been made using the recommendation contained in Reference 6. This recommendation is that the last source be at least 0.005 ℓ less than the body length. For blunt based bodies with simulated wakes, the last source should be at least 0.005 less than the length of the body plus the wake.

2.5 Modified Geometric Input Data

The Subsonic Source program as well as both of the trajectory programs use body segments along with an associated polynomial for each segment to describe the body shape and the subsonic wake. The polynomial equation which is programmed is:

$$\frac{r}{\ell} = C_1 + C_7 \sqrt{C_2(x/\ell)^2 + C_3(x/\ell) + C_4} + C_5(x/\ell) + C_6(x/\ell)^2 \tag{1}$$

where C_1 through C_7 are the polynomial coefficients which were required input data. Calculation of these coefficients was a cumbersome task which could easily be automated into a geometry subroutine to calculate C_1 through C_7 for a limited number of segment types. By performing these coefficient calculations internally, the geometric input data for each segment is reduced to data which is readily determined from configuration drawings.

Segments of the dispenser missile and submissiles which were used to obtain the "SUBMIS" data were either circular arcs or cylinders and the wakes were assumed to be circular arcs. Another common type of missile segment is a conical frustum. It was decided to limit the current geometry subroutine to three types of segments; these are circular arcs, cylinders, and cones or conical frusta. Equations to define C_1 through C_7 for these three shapes were derived and developed into a subroutine to calculate these coefficients from geometric data which, typically, would be given on a configuration drawing. For an ogive, the general circular arc equation is:

$$(x - x_o)^2 + (r - r_o)^2 = R_o{}^2 \tag{2}$$

Solving this equation for r yields

$$r = r_o \pm \sqrt{R_o{}^2 - (x - x_o)^2} \tag{3}$$

And when Equation (3) is rearranged and put into dimensionless form

$$\frac{r}{\ell} = \frac{r_o}{\ell} \pm \sqrt{-(x/\ell)^2 + 2(x_o/\ell)(x/\ell) + (R_o/\ell)^2 - (x_o/\ell)^2} \tag{4}$$

From a comparison of Equation (4) with (1), it is found that

$$C_1 = r_o/\ell \tag{5}$$

$$C_2 = -1. \tag{6}$$

$$C_3 = 2.(x_o/\ell) \tag{7}$$

$$C_4 = (R_o/\ell)^2 - (x_o/\ell)^2 \tag{8}$$

$$C_5 = 0. \tag{9}$$

$$C_6 = 0 \tag{10}$$

$$C_7 = \pm 1 \text{ [+1 if } r_o \text{ is negative, -1 if } r_o \text{ is positive]} \tag{11}$$

In a similar manner, when a cone or conical frustum is defined by X_I, r_I the coordinates of the upstream end of the segment, and X_F, r_F, the coordinates of the downstream end of the segment, then we have

$$C_1 = \frac{r_I \times X_F - r_F \times X_I}{\ell(X_F - X_I)} \tag{12}$$

$$C_5 = \frac{r_F - r_I}{X_F - X_I} \tag{13}$$

$$C_2 = C_3 = C_4 = C_6 = C_7 = 0 \tag{14}$$

and, for a cylindrical segment

$$C_1 = r_{cyL}/\ell \tag{15}$$

$$C_2 = C_3 = C_4 = C_5 = C_6 = C_7 = 0 \tag{16}$$

These are the three options for segment types currently available in SUBROUTINE GEOMET. Since the coefficient calculations are all handled in this subroutine, it would be quite easy to add options for other types of segments if necessary.

2.6 Program Modifications

As noted above, the input data have been changed to simplify the task of running the program. The original program was limited to bodies with seven or fewer body segments. When consideration was given to developing solutions for the dispenser missile with three open submissile bays, seven segments did not seem to be sufficient to simulate the bounding streamline shapes for the open cavities. Changes were made in the program to permit the use of up to 15 segments to define the total body plus the wake simulation, if required.

The Subsonic Trajectory program requires that the source location and strength for the dispenser missile and the submissile(s) be given as input data. It also requires the polynomial coefficients for the body segments to be supplied as input data. Changes were made to the Subsonic Source program to write all of the body specific geometric data plus the locations and strengths of the sources to a .SRC file which is subsequently read by the Subsonic Trajectory program. These changes in both programs and their allied .CSS file essentially automated the data transfer dramatically and reduced the time and effort required to get a problem ready to run on the Subsonic Trajectory program.

2.7 Program Operation

Appendix A consists of a complete set of tables, computer program listings, and definitions required to operate the Subsonic Source program, to interpret the output and to transfer the necessary files to the

Subsonic Trajectory program. Appendix A.1 presents a detailed definition of the input data required by the program. In general, the geometric input data is such that it may be obtained from a configuration drawing and then input in any available units. All dimensional data must be in the same units. Input files to run this program have been created for the dispenser missile with covered submissile bays and the five submissiles used in the wind tunnel tests A and B for the "SUBMIS" data base. Listings of these input data files are presented in Appendix A.2. The program has been successfully run with all of these files and the location of the first source and the source spacing criteria which are listed resulted from a number of runs with the basic data to optimize these two values for best body fit.

The program performs all the geometric calculations for either submissile or dispenser missile in a dimensionless coordinate system based upon the actual vehicle length, ℓ, to be consistent with the requirements of the Subsonic Trajectory program. A program output file for user evaluation of the body shape compliance and other solution criteria is exactly the same as the original program of Reference 6. An example of this output file for the dispenser missile with covered submissile bays is presented in Appendix A.3. This output file was created when the program was run with the DISPENS1.INP file. The output is arranged in sections with the first section being basically a listing of the input data, although now the polynomial coefficients are calculated from geometric input data. The second section lists the source axial locations (x/ℓ) and body radius (r/ℓ) and surface slopes. This set of data gives the user a means to check all of the geometrical calculations. The third section lists the source locations, x/ℓ and strengths, Q, which will later be transferred as input data to the Subsonic Trajectory program. The fourth section compares the body radius calculated from the source distribution with that calculated from the polynomial. This last section is the data the user evaluates to modify the first source location and source spacing to improve the body fit.

The program also creates an output file for data transfer to the Subsonic Trajectory program. An example of this output file for the dispenser missile is shown in Appendix A.4. This file transfers all fuselage geometric data as well as the number of sources (NSORC) and the location and strengths of the sources. With the Subsonic Trajectory program configured to read this file, the volume of manually input data is minimized since these files are also created for the submissile.

Appendix A.5 is a FORTRAN listing of the Subsonic Source program. With the current program structure, there is a limit of 100 sources. If the selected source spacing results in more than 100 points, a message is written to the output file and the program proceeds to a new case. The output file is written to logical unit 6 and the source output file is written to logical unit 7. Polynomial coefficient calculation occurs in Subroutine GEOMET and if there is ever a requirement to allow for more types of segments, the coding should be added to this subroutine.

The CSS file to run the program on the Perkin-Elmer operating system is presented in Appendix A.6. Since it was usually necessary to run the program several times to determine a first source location and source spacing that resulted in a good fit between the source body and polynomial body, the CSS file was set to delete old files and give the new files the generic "NEARSOR" name. However, this requires the user to rename the NEARSOR.SRC file in order to save it. If a user has a requirement to generate a large data base with the Subsonic Trajectory program which will require a number of vehicle .SRC files, probably a more efficient and easy to use CSS file could be developed. The important thing to remember is that the output data is written to logical unit 6 which should be assigned to an output file or to the printer and that the source data for the Subsonic Trajectory program is written to logical unit number 7 which must be assigned to a file with the .SRC extension.

3.0 SUBSONIC AND SUPERSONIC FIN LIFT CURVE SLOPE

The appropriate methods for determining the submissile fin lift curve slope are discussed in References 4 and 6 for the supersonic and subsonic programs. The subsonic program requires only the basic fin alone lift curve slope while the supersonic program modifies the basic fin lift curve slope with a factor calculated from fin lift carry over on the body and from body lift carry over on the fin. In both cases, there is the requirement to determine the basic fin alone lift curve slope. A survey of some of the available methods, approaches, and techniques indicated that the solution of Reference 8 was as sophisticated as other elements in the submissile aerodynamics solutions. Furthermore, this solution was available on the Perkin-Elmer in subroutine form which could readily be adapted to both trajectory programs.

The fin lift curve slope solution of Reference 8 was based on the data contained in the charts of Reference 9. These charts contain wing or fin data for both subsonic and supersonic speeds for a wide range of fin shape parameters. Application of this method required the incorporation of the subroutines BRIT and TABLE from the computer program of Reference 8 into both of the trajectory programs. Data for the fin lift curve slope and center of pressure from the British Data Sheets are contained in subroutine BRIT as block data and are functions of three variables. TABLE is an interpolation subroutine for functions of three independent variables.

An example of a data set is shown in Figure 1 which is reproduced from Figure 1 of Reference 8. These are subsonic and supersonic data sets. Each set is a function of the reduced aspect ratio, $A\sqrt{1 - M^2}$ or $A\sqrt{M^2 - 1}$, and a midchord sweep angle function, $A \tan \Lambda_{1/2}$. This particular chart applies to wings or fins which have a taper ratio, $\lambda = C_t/C_r$, of 0.

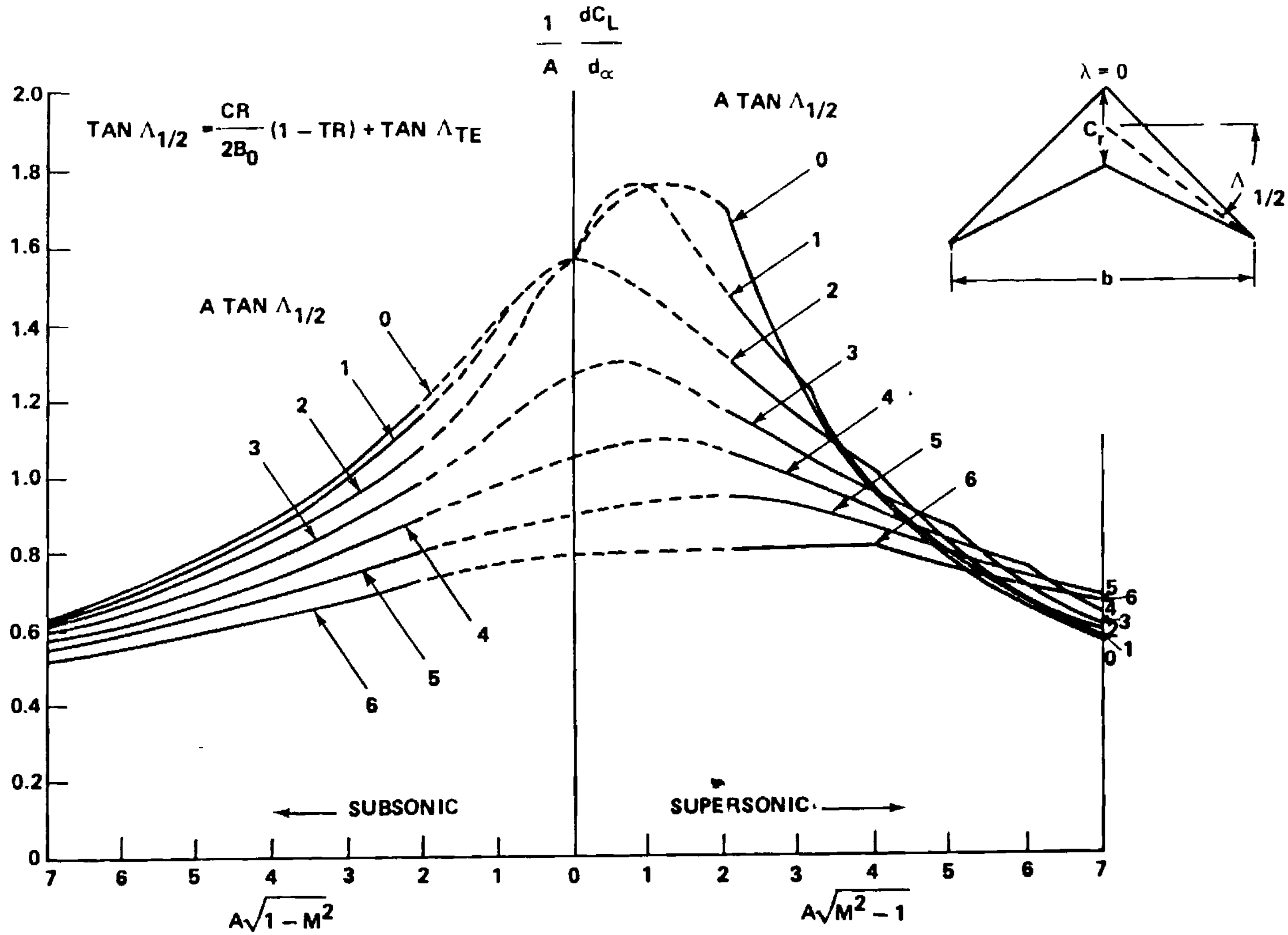

Figure 1. Theoretical slope of lift curve for wings at high speed.

Reference 9 and the computer program subroutine BRIT of Reference 8 contain data for fin taper ratios from 1.0 to 0.0 covering the same range of parameters as shown in Figure 1. Data in this format is available for both the fin lift curve slope, $\frac{1}{A}\frac{dC_L}{d\alpha}$ and the fin center f pressure in percent of the fin mean aerodynamic chord aft of the leading edge of the mean aerodynamic chord.

The procedure for applying the fin alone lift curve slope to the fin in the presence of the body lift curve slope for supersonic speeds was defined, in general, in Appendix II of Reference 4. Specific application of this procedure for rectangular fins was presented in Reference 1. Restriction to rectangular fins was acceptable for the basic purposes of Reference 1. However, in the present case, it is desirable to have the capability of calculating the required fin interference quantities for any fin shape. Equation (1) of Reference 1 defines the required fin lift curve slope as:

$$C_{L_\alpha} = C_{L_{\alpha F}}\left(\frac{S_F}{S_R}\right)\left[\frac{K_{F(B)SB} + K_{B(F)LT}}{K_{F(B)SB} + K_{B(F)SB}}\right] \tag{17}$$

and the K's as defined in the reference are the ratios of the lift of the fin or body in the presence of the body or fin to the lift of the fin alone. Equations defining these ratios may be found in Reference 10. Slender body lift ratios are independent of fin shape and depend only upon the ratio of fin semi-span to body radius and are given in Reference 1 as Equations (5) and (6). Now $K_{B(F)LT}$, the ratio of body carry over lift due to the fin divided by the fin alone lift, is calculated with linear theory in the present analysis. On the basis of the linear theory, this lift factor is defined by four different equations depending upon whether the fin leading edge is subsonic or supersonic and whether or not there is a fuselage afterbody behind the fin trailing edge. Since these equations are now an integral part of the fin lift calculation in the supersonic trajectory program, they are included herein from Reference 10.

For the case of a long afterbody behind the fin trailing edge which allows the total lift carry over to be developed, the lift ratio is defined by Equations (24) and (26) of Reference 10. For the supersonic leading edge case where $\beta m > 1.$, there is (Equation (24), Reference 10):

$$
\begin{aligned}
&K_{B(F)}(\beta C_{L_{\alpha F}})(1+\lambda)\left(\frac{s}{r}-1\right) = \\
&\frac{8\beta m}{\pi\sqrt{\beta^2 m^2 - 1}\left(\frac{\beta d}{c_r}\right)}\left\{\left(\frac{\beta m}{1+\beta m}\right)\left[\frac{(\beta m + 1)\left(\frac{\beta d}{c_r}\right) + \beta m}{\beta m}\right]^2 \cos^{-1}\right. \\
&\left[\frac{1 + (1+\beta m)\frac{\beta d}{c_r}}{\beta m + (\beta m + 1)\frac{\beta d}{c_r}}\right] + \frac{\sqrt{\beta^2 m^2 - 1}}{(\beta m + 1)}\left[\sqrt{1 + 2\frac{\beta d}{c_r}} - 1\right] \\
&\left. - \frac{\sqrt{\beta^2 m^2 - 1}}{\beta m}\left(\frac{\beta d}{c_r}\right)^2 \cosh^{-1}\left(1 + \frac{c_r}{\beta d}\right) - \frac{\beta m}{1+\beta m}\cos^{-1}\left(\frac{1}{\beta m}\right)\right\}
\end{aligned}
\tag{18}
$$

and for the subsonic leading edge case, $\beta m < 1$, the result is (Equation (26), Reference 10):

$$K_{B(F)}(\beta C_{L_{\alpha F}})(1+\lambda)\left(\frac{s}{r}-1\right)=$$

$$\frac{16\left(\frac{\beta m}{1+m}\right)^2}{\pi\left(\frac{\beta d}{c_r}\right)}\left\{\left[\frac{\beta m+(1+\beta m)\left(\frac{\beta d}{c_r}\right)}{\beta m}\right]^{3/2}+\left[\frac{\beta m+(1+\beta m)\left(\frac{\beta d}{c_r}\right)}{\beta m}\right]^{1/2}\right.$$

$$\left.-2-\left[\frac{(1+\beta m)\left(\frac{\beta d}{c_r}\right)}{\beta_m}\right]^2\tanh^{-1}\sqrt{\frac{\beta m}{\beta m+(1+\beta m)\left(\frac{\beta d}{c_r}\right)}}\right\} \tag{19}$$

The cases for no afterbody behind the fin trailing edge are solved by Equations (30) and (31) of Reference 10. The supersonic leading edge solution, $\beta m > 1$, for this case is (Equation (30), Reference 10):

$$K_{B(F)}(\beta C_{L_{\alpha F}})(1+\lambda)\left(\frac{s}{r}-1\right)=$$

$$\frac{8}{\pi\sqrt{\beta^2 m^2-1}}\left(\frac{\beta d}{c_r}\right)\left[\left(1+\frac{mc_r}{d}\right)^2\cos^{-1}\left(\frac{\beta m+\frac{c_r}{\beta d}}{1+\frac{mc_r}{d}}\right)-m^2\beta^2\left(\frac{c_r}{\beta d}\right)^2\cos^{-1}\left(\frac{1}{\beta m}\right)\right.$$

$$\left.+\beta m\left(\frac{cr}{\beta d}\right)^2\sqrt{\beta^2 m^2-1}\sin^{-1}\left(\frac{\beta d}{cr}\right)-\sqrt{\beta^2 m^2-1}\cosh^{-1}\frac{cr}{\beta d}\right] \tag{20}$$

while, for the subsonic leading edge case, $\beta m < 1$, there is (Equation (31), Reference 10):

$$K_{B(F)}(\beta C_{L_{\alpha_F}})(1+\lambda)\left(\frac{s}{r}-1\right) =$$

$$\frac{16\sqrt{\beta m}}{\pi(\beta m+1)}\left(\frac{\beta d}{c_r}\right)\left\{\left(1+\frac{mc_r}{d}\right)\sqrt{\left(\frac{c_r}{\beta d}-1\right)\left(\frac{mc_r}{d}+1\right)}-\left(\frac{c_r}{\beta d}\right)^2(\beta m)^{3/2}\right.$$

$$+\ \beta m\left(\frac{c_r}{\beta d}\right)^2(\beta m+1)\left[\tan^{-1}\sqrt{\frac{1}{\beta m}}-\tan^{-1}\sqrt{\frac{c_r}{\beta d}-1\left(\frac{mc_r}{\beta d}+1\right)}\right]$$

$$\left.-\ \frac{\beta m+1}{\sqrt{\beta m}}\tanh^{-1}\sqrt{\beta m\left(\frac{c_r}{\beta d}-1\right)\Big/\left(\frac{mc_r}{d}+1\right)}\right\} \tag{21}$$

For a rectangular fin, the co-tangent of the leading edge sweep, m, and, therefore βm also, is infinite. This means that limiting forms of the supersonic leading edge, Equations (18) and (20), must be found as m approaches infinity. The result of this process, for the no afterbody case, was presented in Reference 1 as Equation (7) and (8). To complete the set of possible solutions, a corresponding limiting solution for the afterbody case is required. This equation is:

$$K_{B(F)} = \frac{2}{\pi} \frac{1}{(\beta A - 1/2)} \left\{ \frac{1}{2} \left[\frac{1 + \beta A \frac{r}{s}}{1 - \frac{r}{s}} \right] \cos^{-1} \left[\frac{\beta A}{\beta A + \frac{s}{r} - 1} \right] \right.$$

$$- \frac{1}{2} \left[\frac{\beta A \frac{r}{s}}{1 - \frac{r}{s}} \right]^2 \cosh^{-1} \left[1 + \frac{1 - \frac{r}{s}}{\beta A \frac{r}{s}} \right]$$

$$\left. - \frac{1}{2} - \frac{\pi}{4} + \frac{1}{2} \sqrt{1 + \frac{\beta 2\, A \frac{r}{s}}{1 - \frac{r}{s}}} \right\} \qquad (22)$$

when it is put in the same form as Equation (8) of Reference 1.

There are certain limitations on the application of the above equations to specific situations which are discussed in Reference 10. The subroutine to perform the original rectangular fin data, FINCALC, was revised to include all of the above equations with the necessary logic to achieve the required type of solution. Also, the necessary logic to adhere to the limitations discussed in Reference 10 has been written into the subroutine. With the subroutines FINCALC, BRIT, and TABLE included in the Supersonic Trajectory program, the general capability of calculating the fin lift curve slope and center of pressure exists for any required fin parameters. A logical variable, NEMP, lets the program know whether or not fins are present. If fins are present, geometric data for defining the size, shape, and location of the fins must be input for the calculation of the fin data.

4.0 CROSS-FLOW DRAG COEFFICIENT

An excellent review of the better known empirical viscous cross-flow techniques for bodies of revolution was presented in Reference 11. This review covered the methods of Perkins and Jorgenson (Reference 12), Allen (Reference 13), and Kelly (Reference 14). The basic concept of all the techniques is to relate the non-linear viscous normal force of a body of revolution to the drag coefficient of a circular cylinder. The actual implementation of the concept always requires various approximations and sometimes raises questions. Viscous cross-flow force is created by the boundary layer separation on the leeward side of the body. Separation from an increasing radius nose such as an ogive has been shown to be different from separation on a cylinder. Thus, in relating the body cross flow force to the infinite cylinder, end effects must be accounted for in some manner.

The methods reviewed in Reference 11 took the basic approach of calculating the cross flow force from a cylinder drag coefficient which was modified by an efficiency factor that depended upon the fineness ratio of the body. While the basic approach was the same, three different cylinder drag curves were used which had significant differences at low cross flow Mach numbers. Several of these methods were available on the Perkin-Elmer in the form of library subroutines. Based on the review of the various methods and the requirements of both the Subsonic and Supersonic Trajectory programs, subroutines were integrated into the programs to interpolate the cylinder drag coefficient as a function of the cross flow Mach number and the efficiency factor as a function of the submissile ℓ/D. The basic drag curve used for the interpolation is the best curve fit of the collected data in the literature. The efficiency factor in the subroutines was taken from Reference 15.

Based on the review of the literature, the requirements of the program and a limited comparison with experimental data, the chosen method appears equal to or better than the other methods, therefore, alternate methods were not included in the programs. Both trajectory programs require a user input separation point to initiate the cross flow calculation. This gives the program user another way to control the calculation. However, there is very little basic data which would yield "a priori" a separation criteria. Based on the results to date, no general conclusion can be reached as to whether or not inclusion of the non-linear forces and moments will improve the calculated aerodynamic characteristics.

5.0 SUBSONIC TRAJECTORY PROGRAM

This program has its origin in the program described in References 5 and 6 for store separation from aircraft. For application to submissiles separating from dispenser missiles, many of the aircraft elements such as the wing and pylons were bypassed in the original program by the work of Reference 2. In order to quickly and efficiently generate aerodynamic data to compare with experimental data in the "SUBMIS" data base, an option to perform a single variable sweep was inserted into the trajectory integration subroutine. The single variable sweep allows the submissile to perform an axial, vertical, or angle of attack variation with respect to the dispenser missile while all other parameters are held constant.

Following the work of Reference 2 and, later, Reference 1, a thorough program review was conducted. All unused elements found remaining in the program were deleted and an intensive effort was made to streamline the program operation. During this effort, the subroutines to calculate the fin lift curve slope were integrated into the program. The program originally could accept up to seven polynomial segments to define the dispenser missile shape. This is an insufficient number of segments for defining the streamline shapes for three open submissile bays in a dispenser missile. Program modifications were made so that up to 15 polynomial segments could be used to define the dispenser missile shape. At this point, all of the program input data was from a single, formatted, list-directed file including the source locations and strengths for the dispenser missile and submissiles. Since the source locations and strengths are calculated by the Subsonic Source program, these values would have to be manually entered into the input files. This is a very unsatisfactory way to operate when running a modern computer in an interactive mode. Therefore, an entirely new and different input scheme was devised and introduced into the program. The overall plan is discussed and then user supplied data items are defined in detail.

Basically, the general plan is to use (1) Source files for dispenser missile and submissiles which are output from the Subsonic Source Program, (2) a Namelist file for variables dependent upon the dispenser missile and submissile configurations, flight conditions and program control variables which only change infrequently, and (3) a set of interactive data tailored so that multiple data base type runs can be made with minimum effort by the operator. At each step, the emphasis has been put upon developing an overall system which minimizes the total effort required to obtain solutions for a specified group of problems. When this approach is used, the program is limited to one submissile, as in the 'SUBMIS' data base. Solution with multiple submissiles may be obtained using a formatted input data file. This option will be discussed in a subsequent part of this section.

A prerequisite for any trajectory or sweep solution with the Subsonic Trajectory program is to run the Subsonic Source program for each different shape involved and give each of the source files an appropriate name. With the present operating system, these files must be given a .SRC extension, i.e., FILENAME.SRC, to be compatible with the program. The subroutine SRCINPUT reads the source files and prompts the operator to input each .SRC file at the proper time. The .SRC files contain the body geometry data input which is calculated by the Subsonic Source program as well as the source locations and strengths. This automatic pass through of the polynomial coefficients defining each body segment shape means that the user has to input detailed geometric data only to the Subsonic Source program. An example .SRC file for the data base dispenser missile with a simulated wake was presented in Appendix A.4 and a complete identification of the contents of this file is presented in Appendix B.1. The seven polynomial coefficients on the line with each segment number are the coefficients of Equation (1). They are calculated by the Subsonic Source program from the geometric input data and are used by the Subsonic Trajectory program to calculate the dispenser missile and submissile geometry.

The Namelist input data file is composed of data which changes infrequently in the 'SUBMIS' data base. A Namelist file was chosen for this part of the input data primarily for its free format style and the ease with which the user can verify the contents of the file. A comprehensive definition of the variables in the Namelist input file is presented in Appendix B.2. A sample Namelist file for the dispenser missile and the two-caliber ogive nose submissile configurations of the 'ASUBMIS' data base are presented in Appendix B.3.

The interactive part of the input data is read in the subroutine ACTIVINP. This subroutine writes screen cues for each item of data to be input in a sequential order. There are six items of data which are always required to define the initial conditions and the type of calculation to be performed. After these values are input, cues for additional required data which depend upon the type of calculation to be made appear on the console. The console screen cues are presented in Appendix B.4 as they appear on the console screen along with the read statement for the input value associated with the cue. The cues are sufficiently detailed to satisfactorily define the required item. All subsequent calculations have the interactive input menu driven so that only the variables that change need to be reentered.

Overall program arrangement and operation has been tailored to optimize generation of data bases like the current 'SUBMIS' data base which has one submissile in the presence of a dispenser missile and the submissile performs an α, x, or z sweep. Emphasis has been placed upon developing a straightforward operating system based upon the input arrangement with separate files. However, the program can still be operated with a formatted input file plus the appropriate source files and this program operation will be discussed later. In the multiple file, interactive operation, the program writes out screen cues to input (1) the Namelist file, (2) the dispenser missile Source file, (3) the submissile Source file. Multiple case calculations can be made with the same files without reentering the file names. If different files are required for subsequent cases the appropriate filename can be entered. Also, the interactive input subroutine has been tailored so that any case after the first can be run by entering only those values which have changed from the previous run. The program has the capability, as an option, to generate binary data files which have the same format as the 'SUBMIS' data base files. After program execution is completed, a screen cue appears for binary data file creation. If the user elects to create the binary data file, the program cue requests a name for this file. A FORTRAN listing of the version of the Subsonic Trajectory program just described is presented in Appendix B.5. The CSS program to execute this program is given in Appendix B.6. These files are available for use on the Perkin-Elmer as NEARSUB.FTN and NEARSUB.CSS along with the other files necessary for program execution.

The original program of Reference 6 has the capability of obtaining solutions to problems with up to ten stores (submissiles) present. However, only one of the stores was allowed to move with respect to the aircraft (dispenser missile). This capability does not really solve the multiple submissile dispense problem; however, when properly used with the trajectory restart procedure an approximate solution to this type of problem can be obtained. Therefore, this capability has been retained in the program. The multiple submissile cases must be exercised with a formatted input file since the interactive input subroutine was not configured to input the data required for more than one submissile. A sequential listing of the required input data for the program along with the format and detailed definitions is presented in Appendix B.7. When the program is executed for multiple submissiles with input via a formatted file, the source files for the dispenser missile and each of the submissiles are input after a program cue by entering the name of the .SRC file.

The submissile input data is read in a DO LOOP in subroutine STRIO with indices (submissile numbers) varying from 1 to NSTRS (the total number of submissiles). The order in which the formatted input data is read for the various submissiles (Item 6, Appendix B.7) is arbitrary. However, the user must (1) input the submissile .SRC files in the sequence chosen for the formatted data and (2) NEJECT must be the sequence number in the DO LOOP where the input data for the submissile that is to be separated is read. With the majority of the total set of input data for the complete problem being transferred through the .SRC files created by the Subsonic Source program, the formatted input file is very much simpler to create than the original program's formatted input file. An example input file for a dispenser missile with three different submissiles present, with submissile No. 2, the one which separates, is presented in Appendix B.8.

6.0 SUPERSONIC TRAJECTORY PROGRAM

The Supersonic Trajectory program originates from the program described in References 3 and 4 which was developed for the supersonic separation of a store from an aircraft. As was the case in the Subsonic Trajectory program, there were aircraft elements which were not needed for the supersonic submissile dispense solution. The present work, starting from the result of Reference 2, with Reference 1 considered, has deleted all of the program elements not needed for the submissile application and all program elements left unused by these deletions have also been deleted. After this thorough program analysis, the cross flow drag coefficient and the fin lift curve slope calculations, discussed in earlier sections, were integrated into the program to eliminate the tedious calculations required to obtain these data for input values. Also, the single variable sweep capability was integrated into the integration subroutine as an optional type calculation for efficient generation of analytical data to compare with the 'SUBMIS' data base experimental data. The definition of the complete dispenser missile is by segments which are defined by the polynomial Equation (1), and in the original program the body could have up to seven segments. The dispenser missile used for the 'SUBMIS' data base tests has three open submissile bays. Calculations for open bay configurations require a simulation of the free streamline shape around the open bays. Seven segments would not be a sufficient number to define such a complex configuration. Therefore, modifications were made to the program to use up to 15 segments to define the dispenser missile body shape.

One of the most important objectives of the current effort was to create a program which was easy for the user to operate and had the capability of performing a submissile separation trajectory or calculating submissile aerodynamic data for a single variable sweep. This objective was achieved by inserting additional calculations into the program to minimize input data preparation time and by providing an alternative input method which minimizes the time and effort required to create the input data file. This new input method is tailored to match the type of runs required to generate parametric data similar to the 'SUBMIS' data. Calculations of the polynomial coefficients for the body segments were programmed into subroutine GEOMET for a limited number of segment types (See Section 2.5 for details). This subroutine was integrated into the program along with the required geometry input data, which now can be read from a configuration drawing, and the coefficients were calculated. The combination of the geometry and fin lift curve slope calculations eliminates all of the time consuming hand calculations which were previously required to prepare a set of input data.

The original program required a formatted input file which was somewhat cumbersome to prepare, and which for multiple data base type runs, had to be completely duplicated with perhaps only one input value changed for the next run. As an alternative to the formatted input file, a combination Namelist and interactive input procedure was integrated into the program. The basic advantages of the Namelist file are that it is not formatted and the user can check the data without cross referencing a variable and format list since the variable name is typed into the Namelist file. The split between the Namelist variables and the interactive input was determined from an analysis of the total set of input data and the variables which change most frequently in the data base runs. A compromise between maximizing the number of runs that can be made with a given Namelist file and the minimum amount of interactive data was determined and was used to configure each list of input variables. A list of the Namelist variables and a definition for each is presented in Appendix C.1. The

Supersonic Trajectory program does not require any precursor program such as the Subsonic Trajectory program so that complete geometric data for both the dispenser missile and submissile must be input through the Namelist file. Geometric calculations of the polynomial coefficients and the required input data is identical to that defined for the Subsonic Source program in Section 2.5. The same subroutine for performing these calculations was integrated into both programs and the same input data is required. However, the appearance of this data in the Namelist file is somewhat different than in the unformatted, list-directed read file of the Subsonic Source program. There are several items such as the segment end points or segment types which are one-dimensional arrays that may either be entered as a data string or with indexed symbols followed by the appropriate piece of data. Each different type of section requires a different set of input data and, as the program currently stands, each of these pieces of data is entered with a symbol which is indexed with the segment number.

The interactive data primarily sets the type of calculation to be performed and enters the initial conditions of the submissile with respect to the dispenser missile. Each read in the subroutine ACTIVINP is preceded by a screen cue identifying the data to be entered while giving a description of the required data. A complete listing of the screen cues, the required input data, and some additional information is presented in Appendix C.2. All of this appendix is given in the same sequence as the program requires it to be input so it can also be used as a sequence guide for preparing the interactive input. Since the major criteria for the Namelist/Interactive data split was the ability to efficiently generate data base type data sets where runs frequently differ by a change of only one variable, for all calculations after the first case, a menu type interactive input is used where only the variables which change must be input. The program is still configured so that it can be run from either a complete formatted input file or from the combination of the Namelist and Interactive data. The CSS file which loads and runs the program is structured such that if a formatted input file is to be used, the input file name must be typed in following the program name. When this is not done the program requires a Namelist file, and a screen cue appears after the program is loaded and requests that the filename of the Namelist data file be entered. The Namelist file name must have a .NAM extension as the program is currently structured. A copy of the CSS file is presented in Appendix C.3. Namelist input files for the data base configuration N2D1 and for an open bay dispenser missile with simulated free streamlines are presented in Appendices C.4 and C.5, respectively. The geometric details of configuration N2D1 are documented in References 7 and 1; thus, Appendix C.4 applies to a documented configuration for the user's application. Appendix C.5 illustrates the application to a dispenser missile configuration consisting of more than seven segments with one or more segments of each type that the program handles.

Standard program output is directed to an output file with filename .OPT which always carries the OPT extension. The filename is either the formatted input filename, if it is used, or it is the program filename, if the Namelist input file is used. The standard output file may consist only of summary data at each trajectory or sweep point or it may also contain detailed submissile load distributions at each of these points. A screen cue is presented to the user to enable a choice of the type of output to be made. Load distributions lengthen the output file significantly. A binary output file which is configured to the 'SUBMIS' data base files format is also available as a program option. A screen cue appears to permit the user to select or bypass this option. If the option is selected another cue appears requesting a name for the binary data file to be entered. All calculations made during one program entry are put into the same binary file.

A FORTRAN listing of the final version of the program is presented in Appendix C.6. The CSS, FORTRAN, and executable files are available on the Perkin-Elmer with the filename NEARSUP with the appropriate extensions. This program will execute with either of the Namelist files in Appendices C.4 and C.5 with an appropriate set of interactive data.

7.0 SAMPLE RESULTS

The Subsonic and Supersonic trajectory programs have been used to calculate results to compare with the experimental data in the 'SUBMIS' data base. Comparisons which will be presented are not meant to be exhaustive; however, the effects of important physical parameters upon both the experimental data and calculated results will be shown. Data base configurations consist of two dispenser missiles which have the same overall shape but have either covered or open submissile bays. There are five differently shaped or sized submissiles, some of them both with and without fins, which were tested in the presence of a dispenser missile. A sketch of the dispenser missile which was used in tests A and B, is presented in Figure 2. The submissiles which were used in test A are shown in Figure 3 along with their data base configuration notations. Sketches of the submissiles used in test B are shown in Figure 4 and are identified with their data base configuration notations. Table 1 presents a complete set of configuration nomenclature.

Source and sink distributions which are used to represent dispenser missile fuselage volume effects are ideally suited for the dispenser missile with covered submissile bays. However, they can only be used to

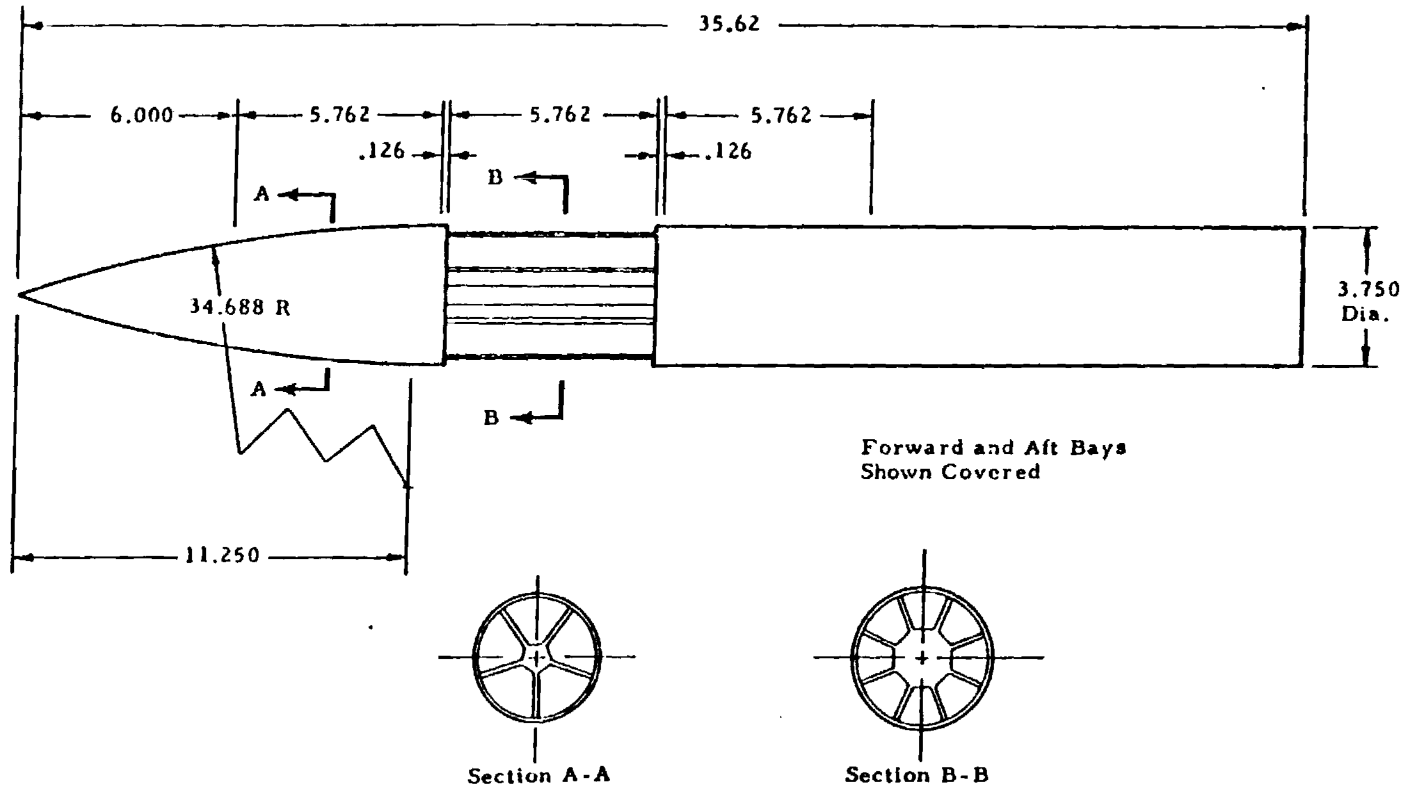

Figure 2. Dispenser missile design.

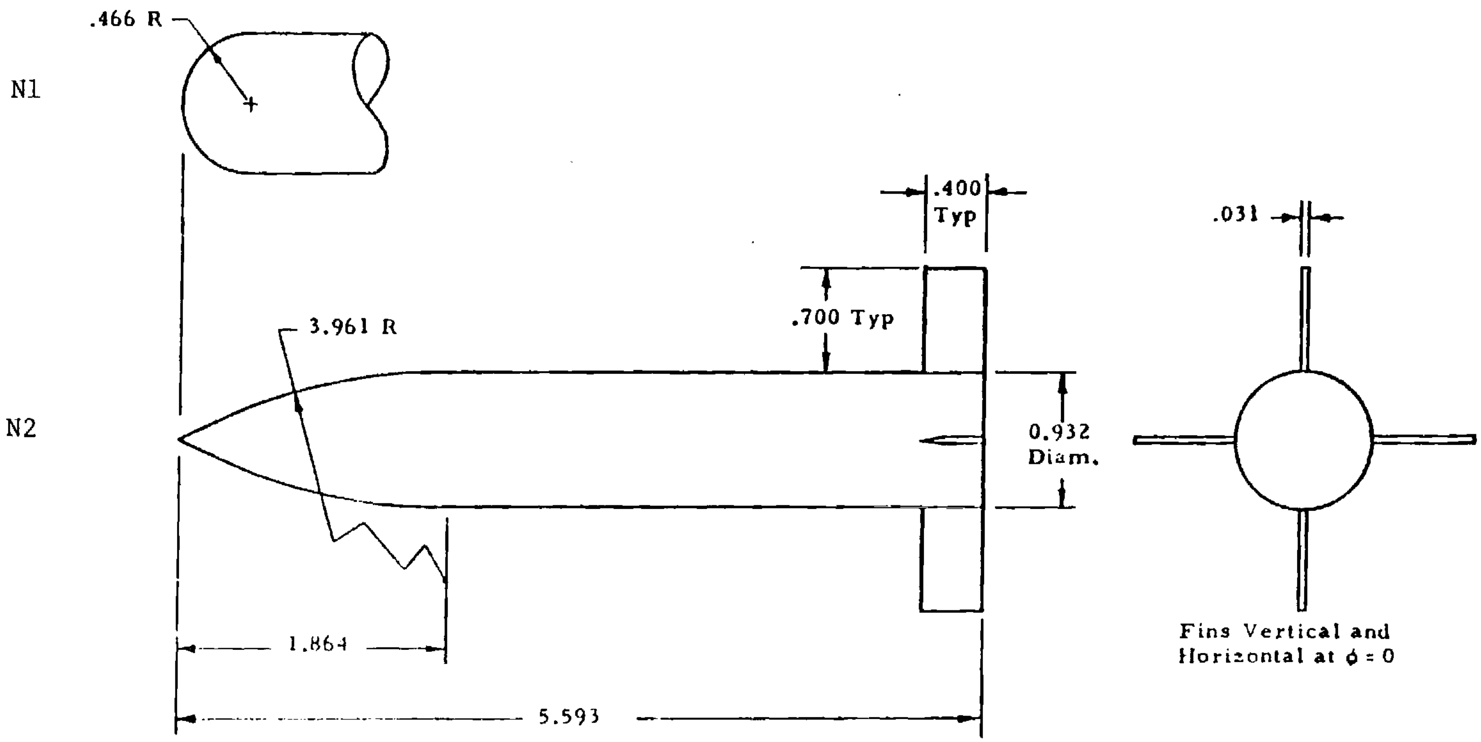

Figure 3. Submissile design for Test A.

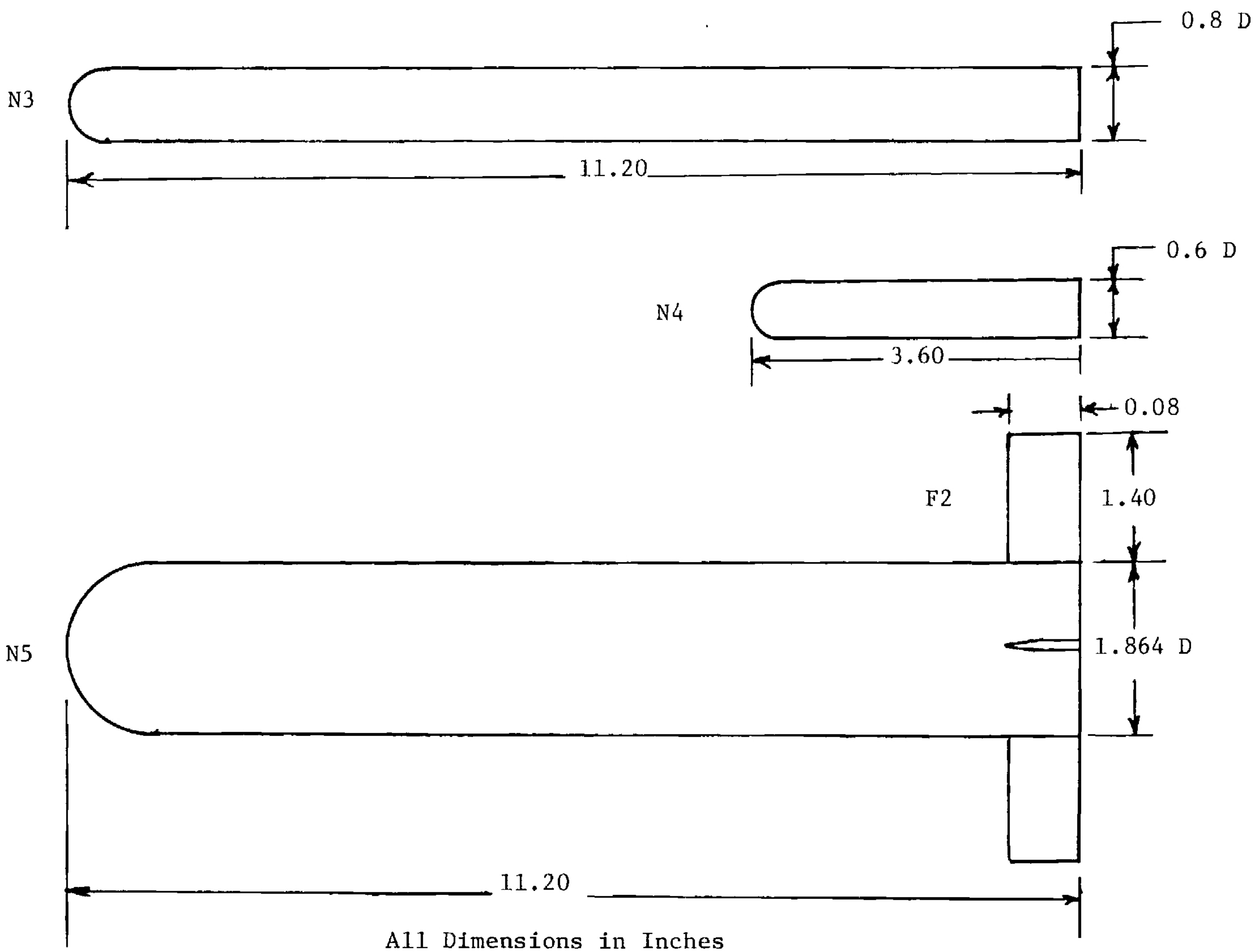

Figure 4. <u>Submissile design for Test B.</u>

TABLE 1. CONFIGURATION NOMENCLATURE

Symbol	Definition
D1	Dispenser missile with covered submissile bays
D2	Dispenser missile with opened submissile bays
F1	Submissile fins on
N1	Submissile with hemisphere nose (one-half caliber ogive) Test A - see Figure 3
N2	Submissile with two caliber ogive nose Test A - see Figure 3
N3	High slenderness ratio submissile Test B - see Figure 4
N4	Small submissile Test B - see Figure 4
N5	Large submissile Test B - see Figure 4

A complete test configuration consists of up to six alphanumeric characters identifying each element present in the wind tunnel during a run, e.g. N2F1D1. This is the two caliber ogive nose submissile with fins tested with the covered bay dispenser missile.

represent the open bay configuration if a mean streamline bounding the cavity flow is known. Since this type of data is not, in general, available for axi-symmetric cavities, calculations presented herein will be restricted to the dispenser missile with covered submissile bays. Results to be presented are for (1) supersonic speeds and (2) subsonic speeds. Within each of these groups results are presented to define the effects of (a) submissile nose shape, (b) submissile axial location, and (c) dispenser missile angle of attack. The experimental data which will be presented is taken from the 'ASUBMIS' data base and it is always identified by a run number which is Axxxx. Corresponding analytical results are identified with the same four digits but with a different letter of the alphabet, e.g. Hxxxx. All data plots were made with the data base plot program, and Table 2 presents the plot program parameter nomenclature for all of the parameters used herein.

Normal force and pitching moment coefficients for the submissile N1 at a Mach number of 1.2 are presented in Figures 5 and 6 as a function of the vertical separation distance below the center bay. Data is shown for the submissile at 0, +10, and -10 degree angles of attack. The data for both C_N and C_M are well matched at zero angle of attack indicating that the predicted interference effects of the dispenser missile are good. For large angles of attack, the C_N is not well predicted while the C_M is, on the whole, quite well predicted. This mismatch of agreements is probably more of a shortcoming of the slender body theory used to predict submissile aerodynamics than it is of the linear theory used to predict dispenser missile interference velocities. Experimental data and calculations for both C_N and C_M are somewhat asymmetrical at large separation distances. This effect apparently is caused by the reversed tail/nose position in the dispenser missile flow field gradient and it is gratifying that the analysis predicts this, at least qualitatively. However, for the submissile positive angle of attack which is nose up toward the submissile, there is a significant experimental trend at small separation distances which is not

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1125	N1D1	5.00	1.20	3.95	0.00	0.10
*	H1125	N1D1	5.00	1.20	3.95	0.00	0.00
O	A2125	N1D1	5.00	1.20	3.95	9.97	0.10
X	H2125	N1D1	5.00	1.20	3.95	10.00	0.00
□	A3125	N1D1	5.00	1.20	3.95	-10.00	0.10
◇	H3125	N1D1	5.00	1.20	3.95	-10.00	0.00

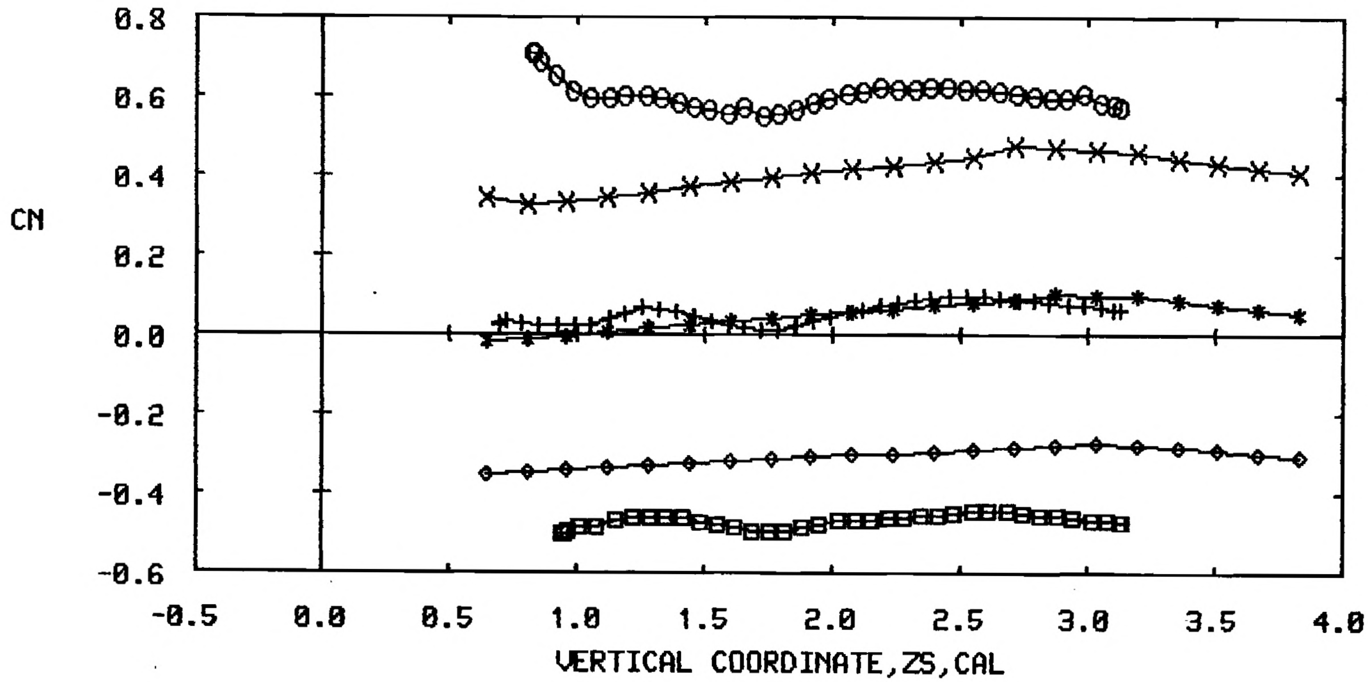

Figure 5. Comparison of normal force coefficients for the submissile, body alone, N1, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1125	N1D1	5.00	1.20	3.95	0.00	0.10
*	H1125	N1D1	5.00	1.20	3.95	0.00	0.00
O	A2125	N1D1	5.00	1.20	3.95	9.97	0.10
X	H2125	N1D1	5.00	1.20	3.95	10.00	0.00
□	A3125	N1D1	5.00	1.20	3.95	-10.00	0.10
◇	H3125	N1D1	5.00	1.20	3.95	-10.00	0.00

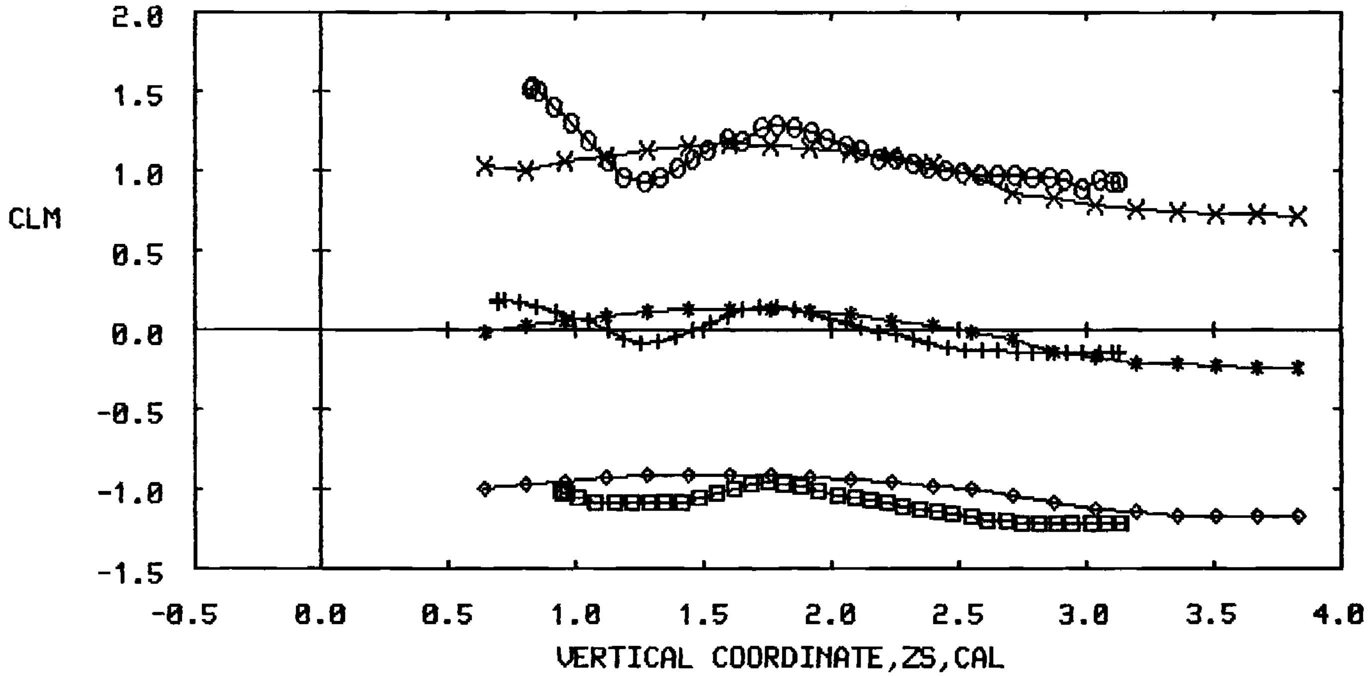

Figure 6. Comparison of pitching moment coefficients for the submissile body alone, N1, in the presence of the dispenser missile, D1, with the relative vertical location varying.

TABLE 2. PLOT PARAMETER NOMENCLATURE

ALPD	Dispenser angle of attack, degrees
ALPS	Submissile angle of attack, degrees
MACH	Free stream mach number
MODE	Type of calculation 1 = angle of attack sweep 4 = longitudinal sweep 5 = vertical sweep
PHIS	Submissile roll angle, degrees
PSIS	Submissile yaw angle, degrees
XS	Longitudinal coordinate, X distance from dispenser missile nose to submissile center of gravity in dispenser calibers, positive aft
ZS	Vertical coordinate, Z distance from dispenser missile center line to submissile center of gravity in dispenser calibers, positive down
CLM	Submissile pitching moment coefficient
CN	Submissile normal force coefficient

predicted by the analysis. This effect is probably due to mutual interference created by the submissile bow shock impinging upon the dispenser missile and being reflected back to the submissile. Such phenomena are not considered by the analysis. The calculations presented in these figures did not include any non-linear cross flow force contribution to C_N and C_M. Including this contribution would improve the C_N agreement but it would degrade the C_M agreement. Figure 7 presents the zero angle of attack pitching moment coefficient of Figure 6 on a greatly expanded scale. This figure also shows that there are significant data trends at small separation distances which are not predicted by the analysis. However, the data and the analytical results agree well at the larger separation distances.

Results presented in the preceeding three figures were data for the hemispherical nose submissile and the results presented in Figures 8 through 10 present a similar set of comparisons for the submissile with the two caliber ogive nose. The general comparison between data and analysis for this submissile is basically the same as for the blunt nose submissile. However, there are subtle differences in the two sets of experimental data at small separation distances which probably represent the change in the mutual interference due to the different bow shock wave shapes of the two submissiles. Thus, the change of the submissile nose shape does not have an important effect upon either the data or the analytical results. Pitching moment coefficients for ten degrees angle of attack from Figure 9 are shown on an expanded scale in Figure 11 in order to better illustrate that, while some qualitative trends are not well predicted, the average value over the separation distance is quite well predicted.

Results which have been discussed above are for the two submissiles separating vertically from the dispenser missile at the position of the center submissile bay. This position is below the cylindrical section of the dispenser missile where one expects small streamline curvature, and therefore, small induced normal velocities. This expectation is well

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
O	A1125	N1D1	5.00	1.20	3.95	0.00	0.10
X	H1125	N1D1	5.00	1.20	3.95	0.00	0.00

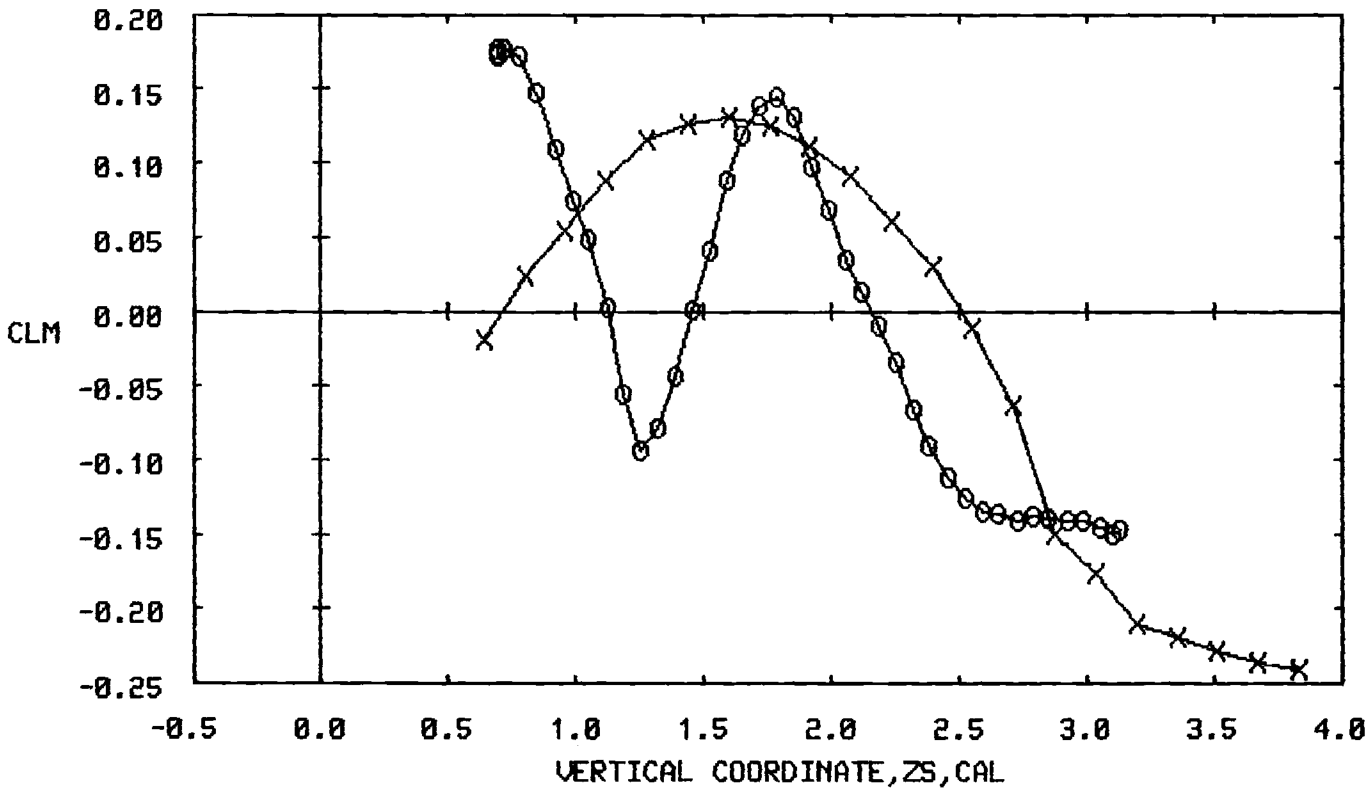

Figure 7. Comparison of pitching moment coefficients for the submissile body alone, N1, in the presence of the dispenser missile, D1, with the relative vertical location varying.

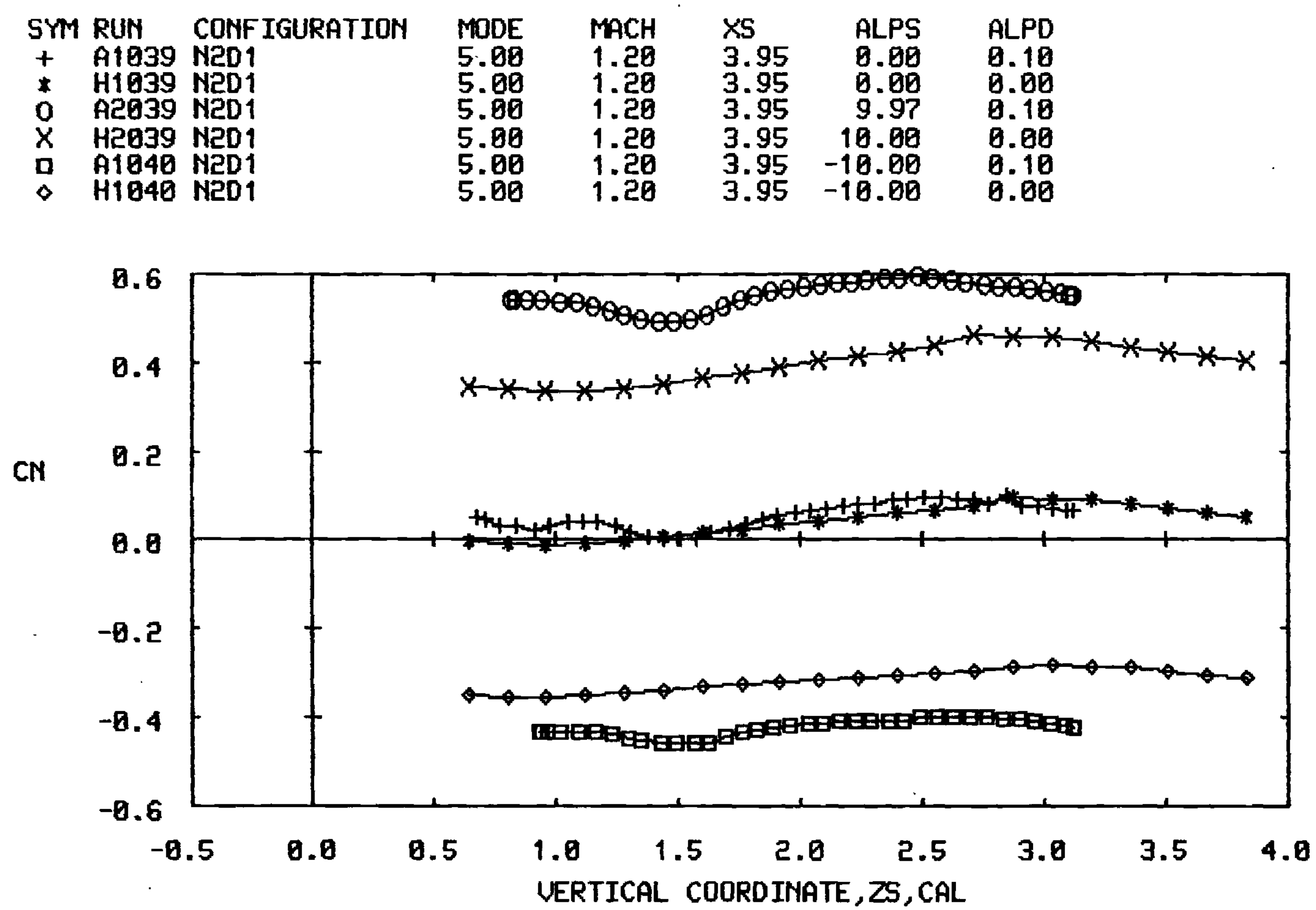

Figure 8. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

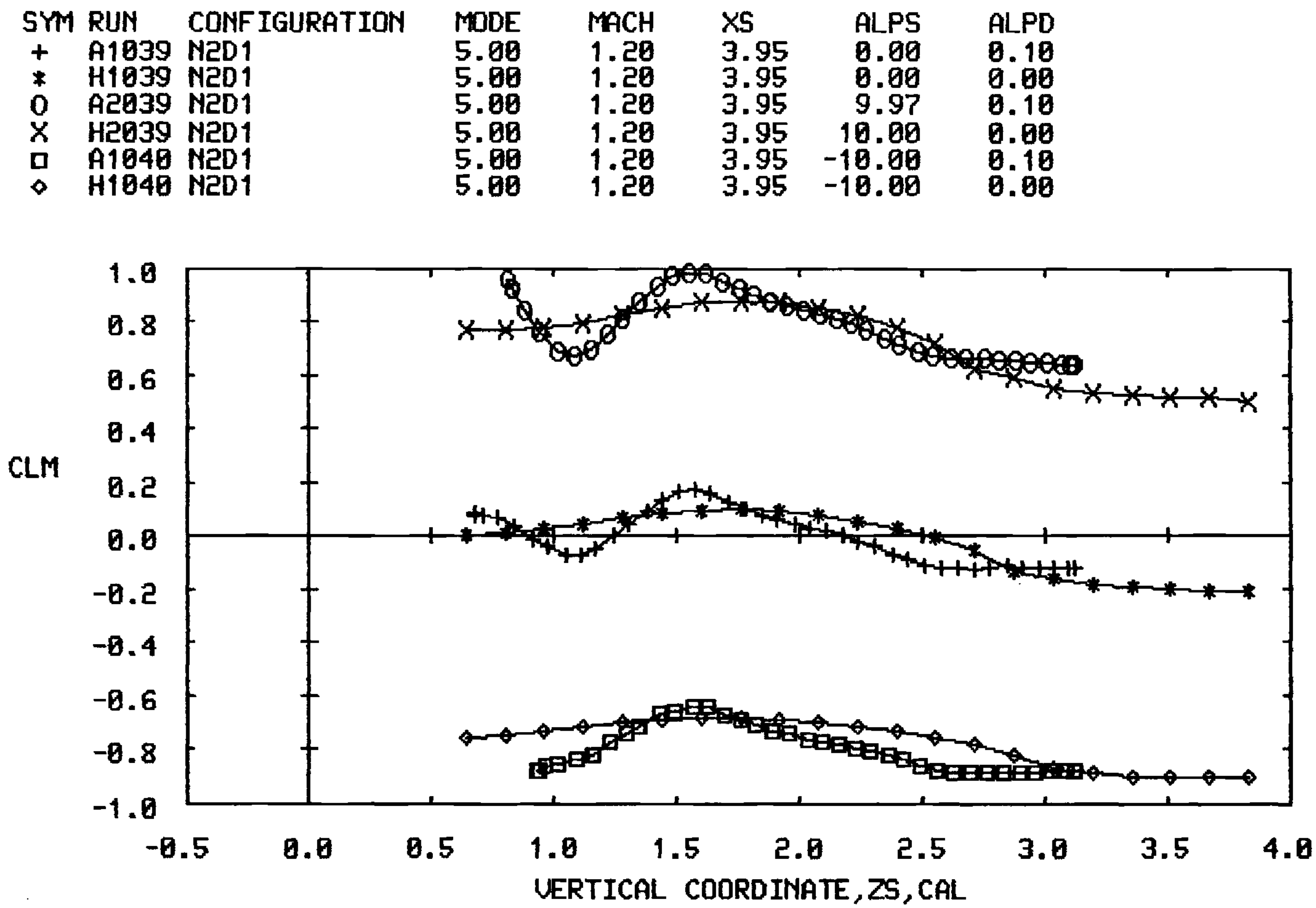

Figure 9. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1039	N2D1	5.00	1.20	3.95	0.00	0.10
*	H1039	N2D1	5.00	1.20	3.95	0.00	0.00

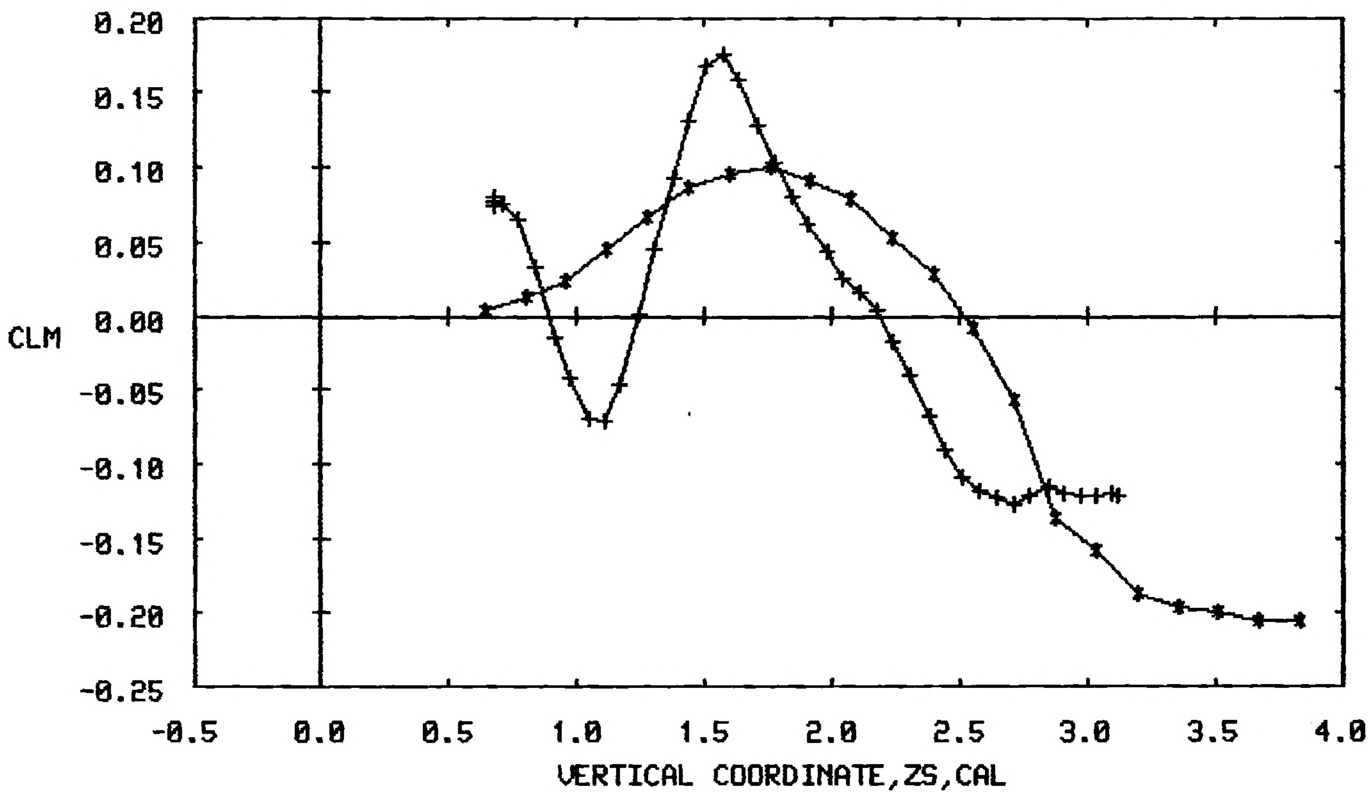

Figure 10. Comparison of pitching moment coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
O	A2039	N2D1	5.00	1.20	3.95	9.97	0.10
X	H2039	N2D1	5.00	1.20	3.95	10.00	0.00

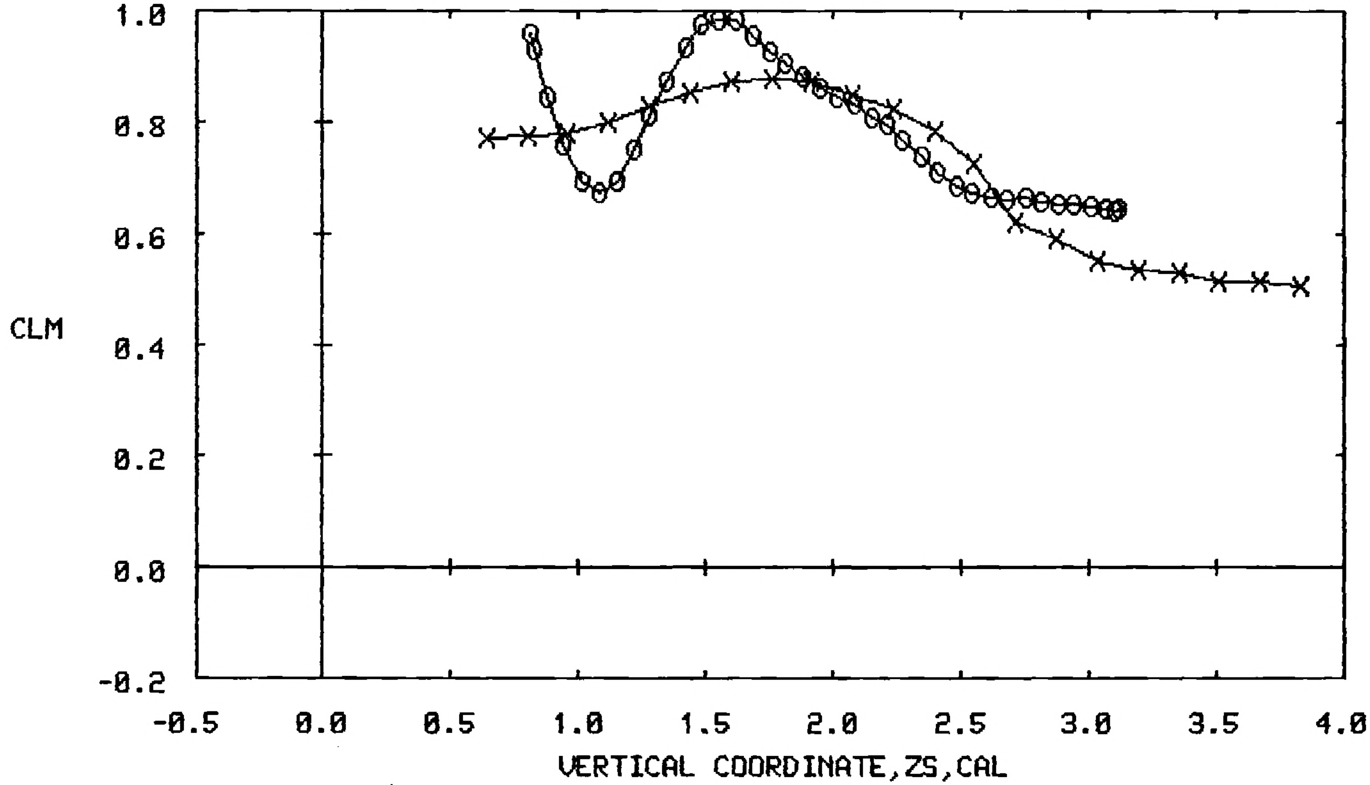

Figure 11. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

verified by the small coefficients seen in the previous figures for submissile angle of attack equal to zero. Figures 13 through 14 present a similar set of data for the two caliber ogive nose submissile separating from the dispenser missile at the position of the front submissile bay. This bay is located forward on the dispenser missile ogive nose where significant streamline curvature is expected. This effect is readily apparent in Figures 12 and 13 where the coefficient slopes are significant over the major portion of the vertical distance. Again, the analysis predicts the qualitative trends of the data very well, which shows that the source/sink distribution adequately simulates the dispenser missile flow field. These two figures, compared to Figures 8 and 9 clearly show there are both qualitative and quantitative effects of these separation positions. Figure 14 presents the pitching moment coefficient at zero angle of attack on an expanded scale and it is seen that the analytical prediction agrees with the data very well. This is substantially different from the results of Figures 7 and 10. It reflects the difference between a flow field with a strong gradient and small mutual interaction and a flow field with a weak gradient and large mutual interaction. Figure 15 presents the normal force coefficient for zero angle of attack from Figure 12 on a greatly expanded scale and it is seen that the agreement between data and analysis is just as good as that for the pitching moment coefficient. This data certainly reinforces the conclusion that the source/sink distribution represents the dispenser missile fuselage volume effects very well.

Another physical variable which affects the submissile aerodynamic data during separation is the dispenser missile angle of attack. An example of this effect for the two caliber ogive nose submissile separating from the center bay position is shown in Figures 16 and 17 with the dispenser missile at a 5.0 degree angle of attack. When these two figures are compared to Figures 8 and 9, the effect of the dispenser missile angle of attack is readily seen. The agreement between the analytical results and the experimental data in Figures 16 and 17 is certainly comparable to

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1041	N2D1	5.00	1.20	2.38	0.01	0.10
*	H1041	N2D1	5.00	1.20	2.38	0.00	0.00
O	A2041	N2D1	5.00	1.20	2.38	9.98	0.10
X	H2041	N2D1	5.00	1.20	2.38	10.00	0.00
□	A1042	N2D1	5.00	1.20	2.38	-9.99	0.10
◇	H1042	N2D1	5.00	1.20	2.38	-10.00	0.00

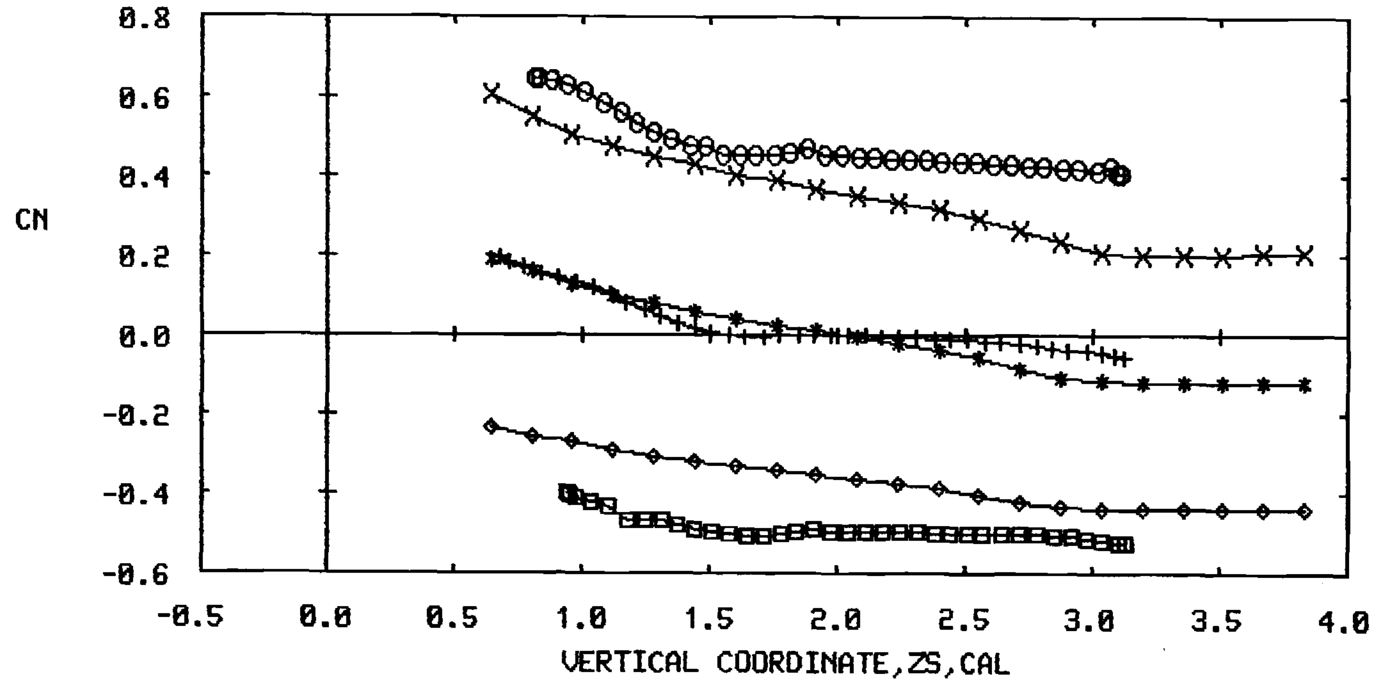

Figure 12. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1041	N2D1	5.00	1.20	2.38	0.01	0.10
*	H1041	N2D1	5.00	1.20	2.38	0.00	0.00
O	A2041	N2D1	5.00	1.20	2.38	9.98	0.10
X	H2041	N2D1	5.00	1.20	2.38	10.00	0.00
□	A1042	N2D1	5.00	1.20	2.38	-9.99	0.10
◇	H1042	N2D1	5.00	1.20	2.38	-10.00	0.00

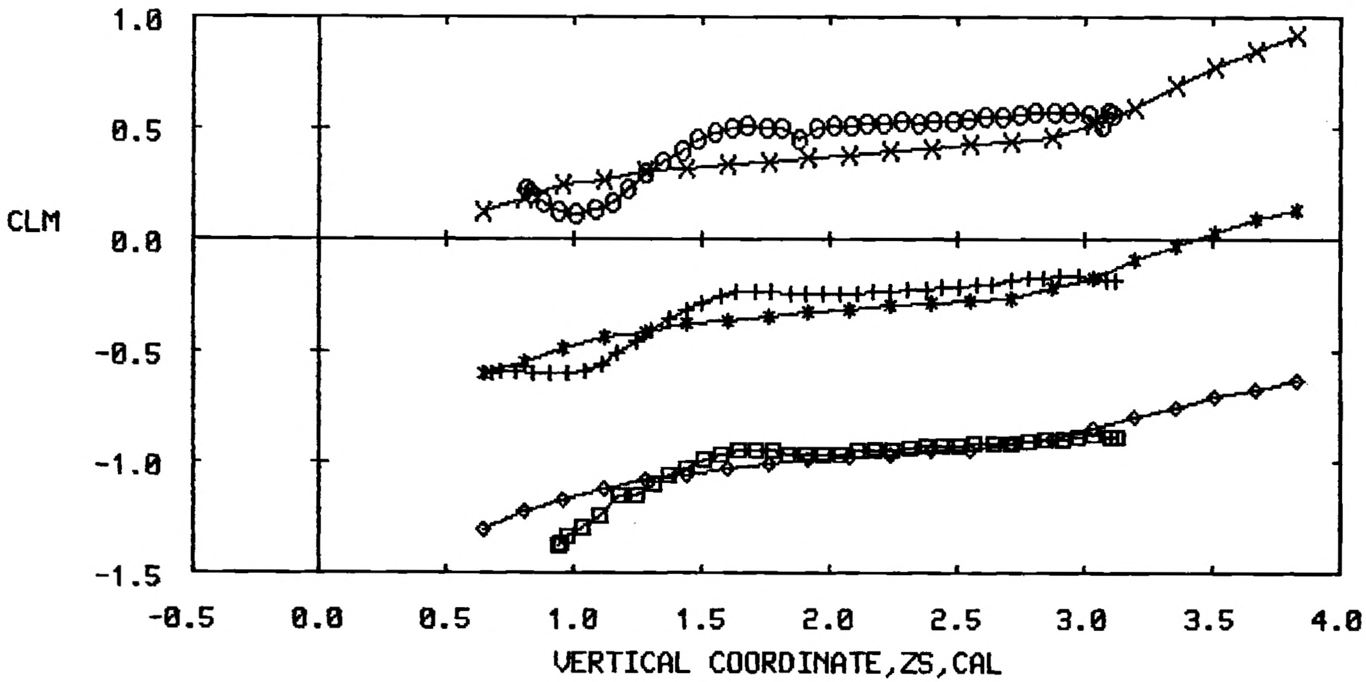

Figure 13. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

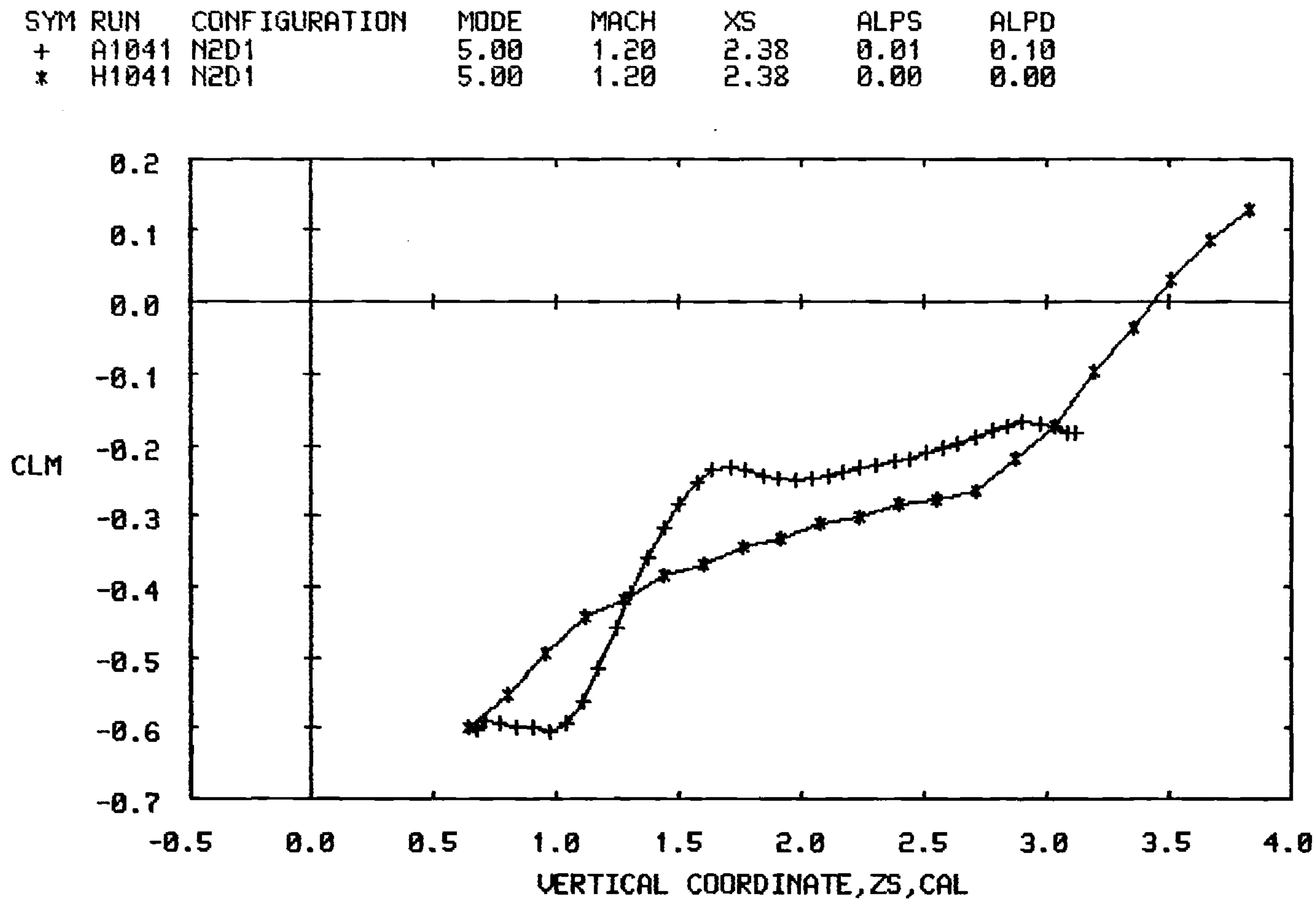

Figure 14. Comparison of pitching moment coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

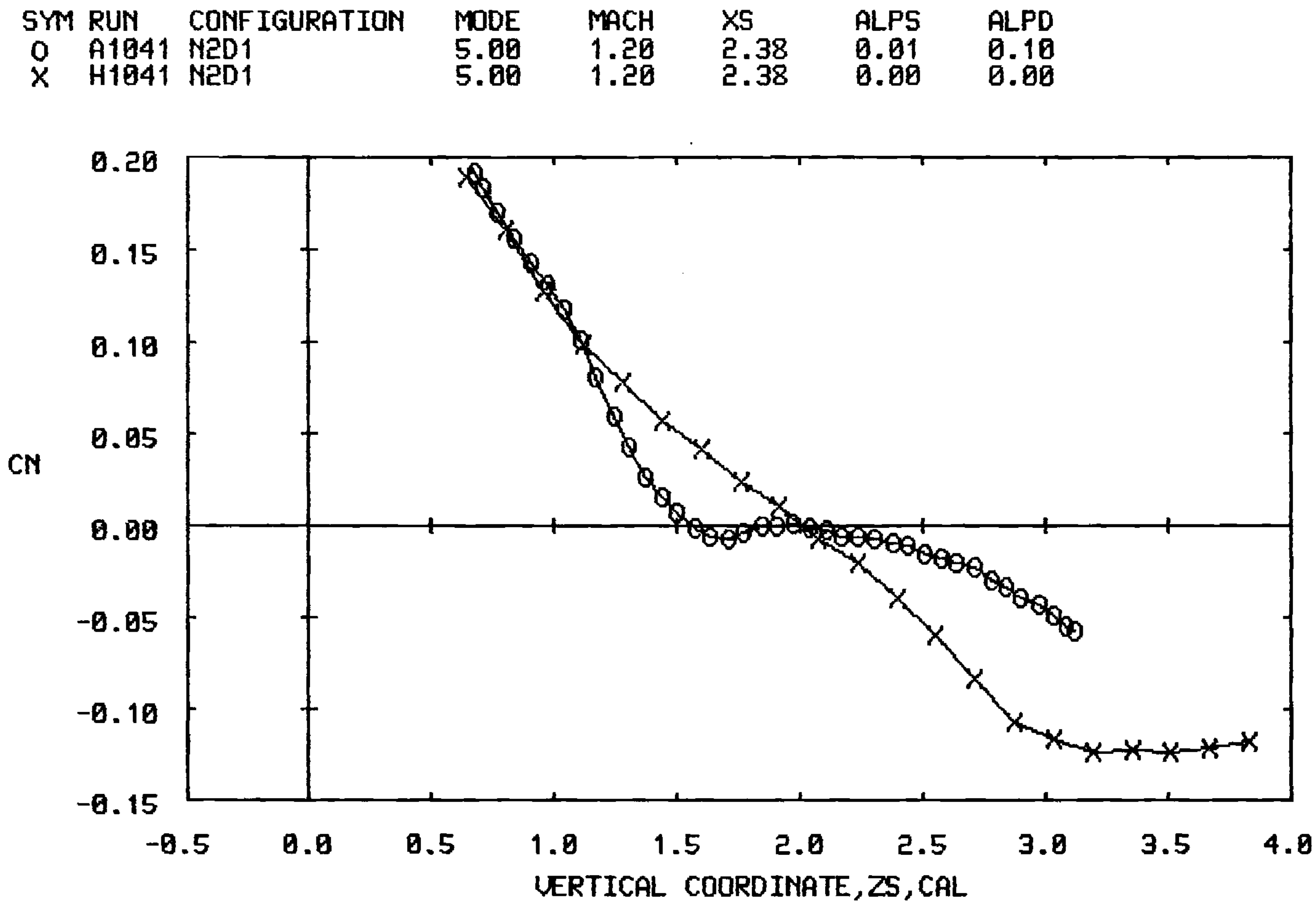

Figure 15. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1095	N2D1	5.00	1.20	3.95	0.00	5.05
*	H1095	N2D1	5.00	1.20	3.95	5.00	5.00
O	A2095	N2D1	5.00	1.20	3.95	10.00	5.04
X	H2095	N2D1	5.00	1.20	3.95	15.00	5.00
□	A3095	N2D1	5.00	1.20	3.95	-9.98	5.04
◇	H3095	N2D1	5.00	1.20	3.95	-5.00	5.00

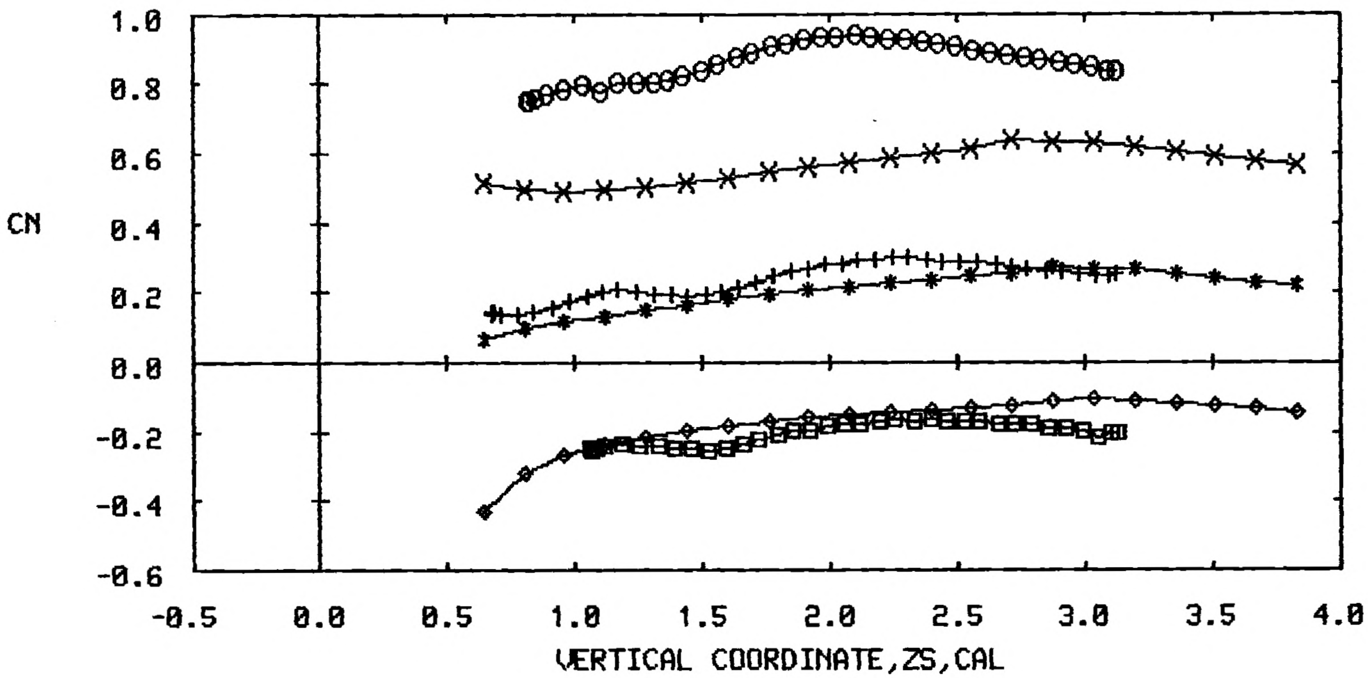

Figure 16. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

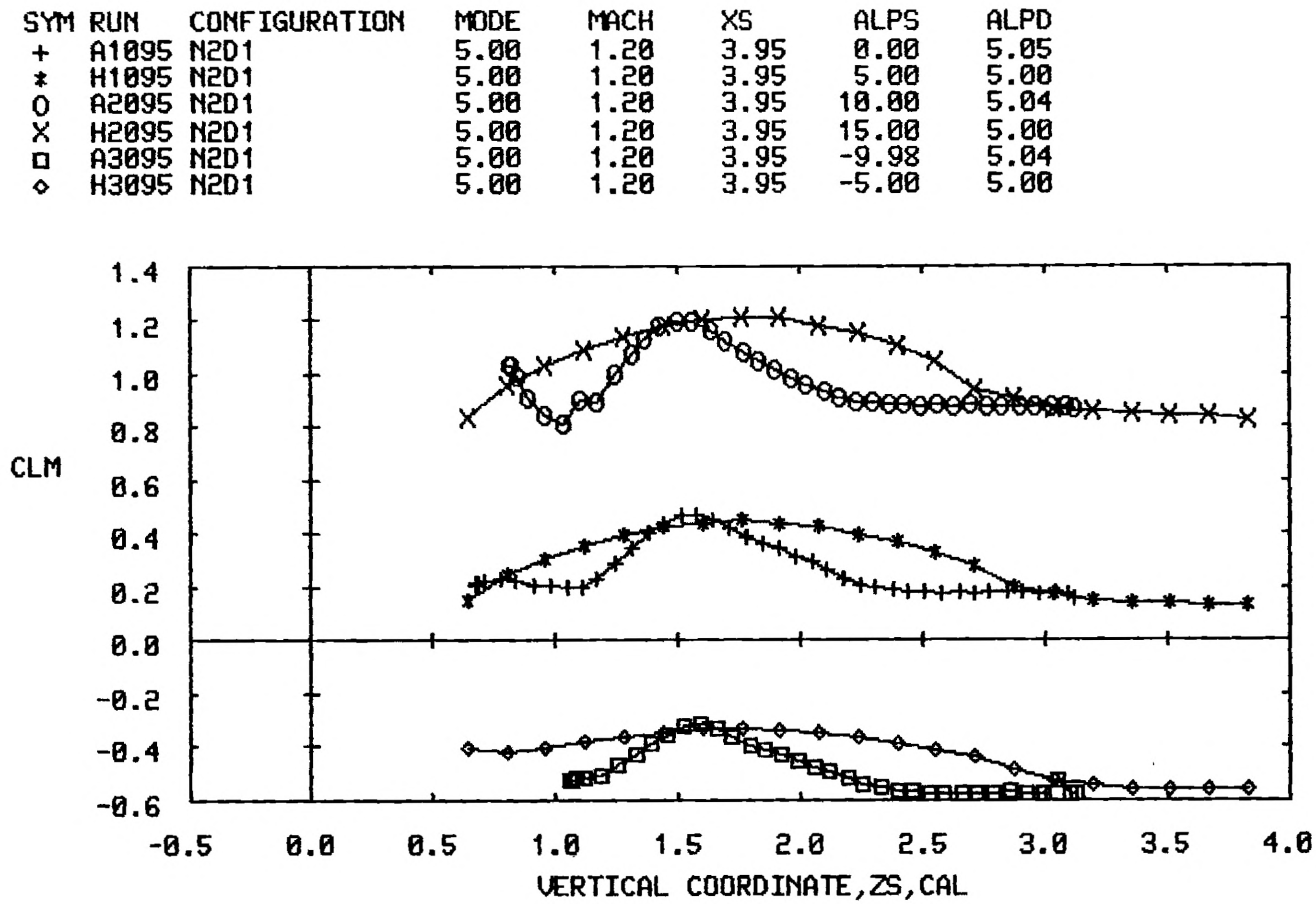

Figure 17. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

the agreement seen in Figures 8 and 9. Thus, it is seen that the doublet distribution is apparently simulating the dispenser missile angle of attack effects very well.

The presentation of the subsonic data and analysis comparisons follows the same basic flow as the supersonic results presented above. Figures 18 and 19 present the normal force and pitching moment coefficients for the submissile N1 at a Mach number of 0.8 as a function of the vertical separation distance from the dispenser missile. The experimental data is independent of separation distance after the submissile reaches one caliber separation while the analytical results are almost completely independent of separation distance. Thus, the source/sink distribution either does not represent the dispenser missile flow field very well or mutual interference is creating significant effects when the two bodies are close together. Slender body theory does a fair job of accounting for submissile angle of attack in this case with reasonable agreement between experimental data and calculated results for angles of attack of $\pm$ ten degrees.

Figures 20 and 21 present the results for a submissile with a two caliber ogive nose for comparison with the two previous figures for the hemispherical nose submissile. The interference effects are very similar for both the experimental data and the analytical results. However, there are significant differences between both experimental data and analytical results for the two configurations at $\pm$ ten degree angles of attack. Further, the analytical results and experimental data do not agree as well for the sharp nose configuration as they do for the blunt nose configuration. The pitching moment coefficients show poor agreement while the normal force coefficient agreement is fair. This is a case where inclusion of non-linear cross flow forces would probably improve the overall agreement. In this case the inclusion of the non-linear normal force and pitching moment due to the cross flow drag would improve the agreement with the data

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1130	N1D1	5.00	0.80	3.95	0.01	0.10
*	I1130	N1D1	5.00	0.80	3.95	0.00	0.00
O	A2130	N1D1	5.00	0.80	3.95	9.99	0.09
X	I2130	N1D1	5.00	0.80	3.95	10.00	0.00
□	A3130	N1D1	5.00	0.80	3.95	-9.98	0.09
◇	I3130	N1D1	5.00	0.80	3.95	-10.00	0.00

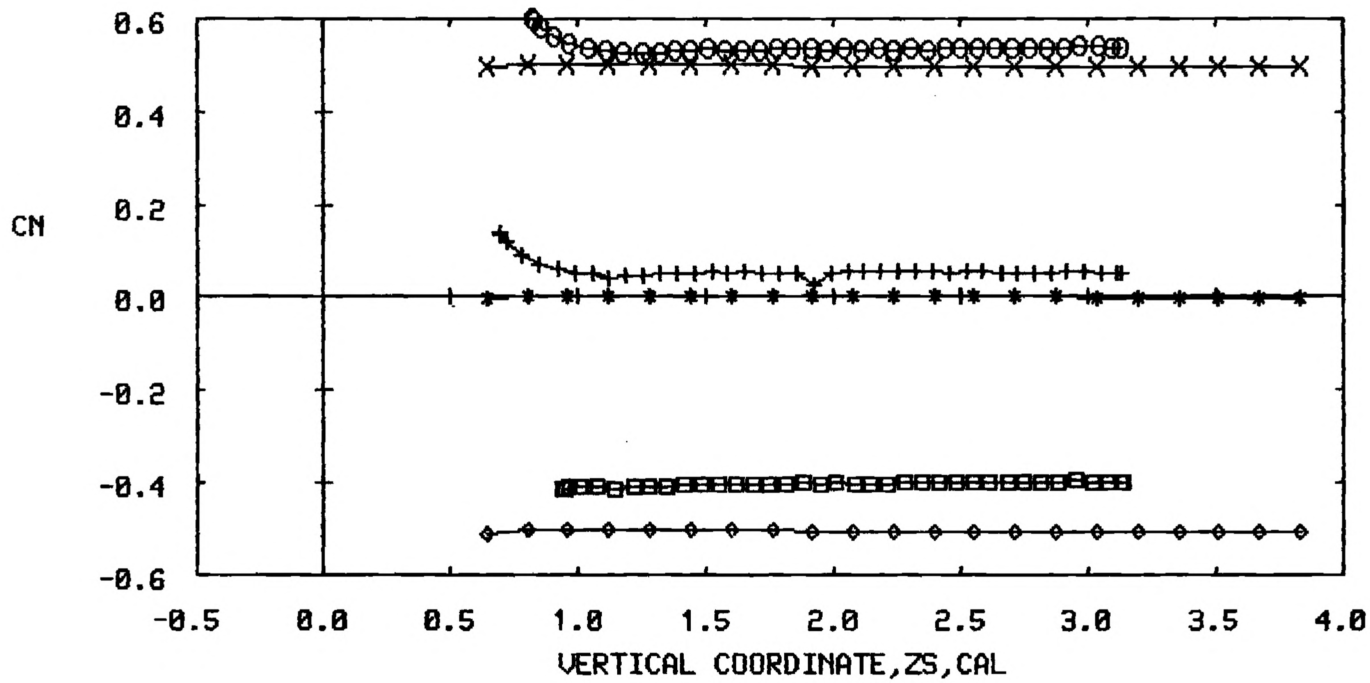

Figure 18. Comparison of normal force coefficients for the submissile, body alone, N1, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1130	N1D1	5.00	0.80	3.95	0.01	0.10
*	I1130	N1D1	5.00	0.80	3.95	0.00	0.00
O	A2130	N1D1	5.00	0.80	3.95	9.99	0.09
X	I2130	N1D1	5.00	0.80	3.95	10.00	0.00
□	A3130	N1D1	5.00	0.80	3.95	-9.98	0.09
◇	I3130	N1D1	5.00	0.80	3.95	-10.00	0.00

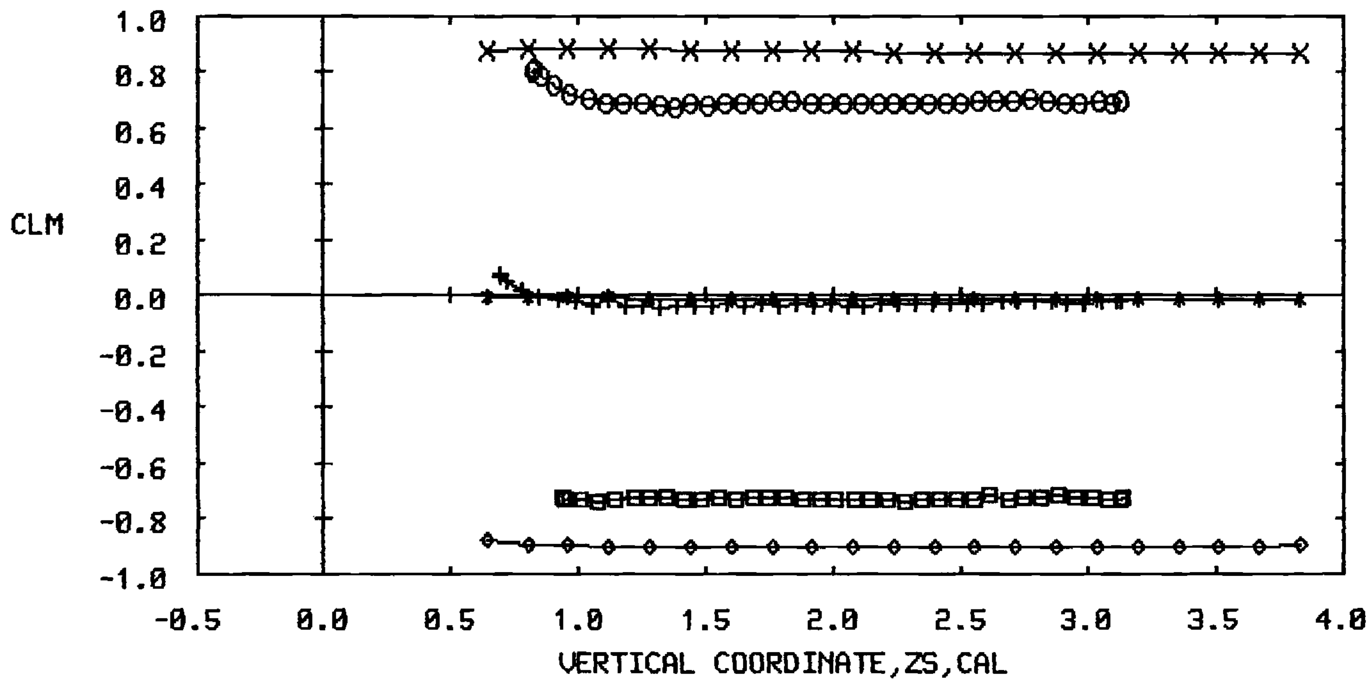

Figure 19. Comparison of pitching moment coefficients for the submissile body alone, N1, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1054	N2D1	5.00	0.80	3.95	0.00	0.10
*	I1054	N2D1	5.00	0.80	3.95	0.00	0.00
O	A2054	N2D1	5.00	0.80	3.95	9.98	0.10
X	I2054	N2D1	5.00	0.80	3.95	10.00	0.00
□	A1055	N2D1	5.00	0.80	3.95	-10.00	0.10
◇	I1055	N2D1	5.00	0.80	3.95	-10.00	0.00

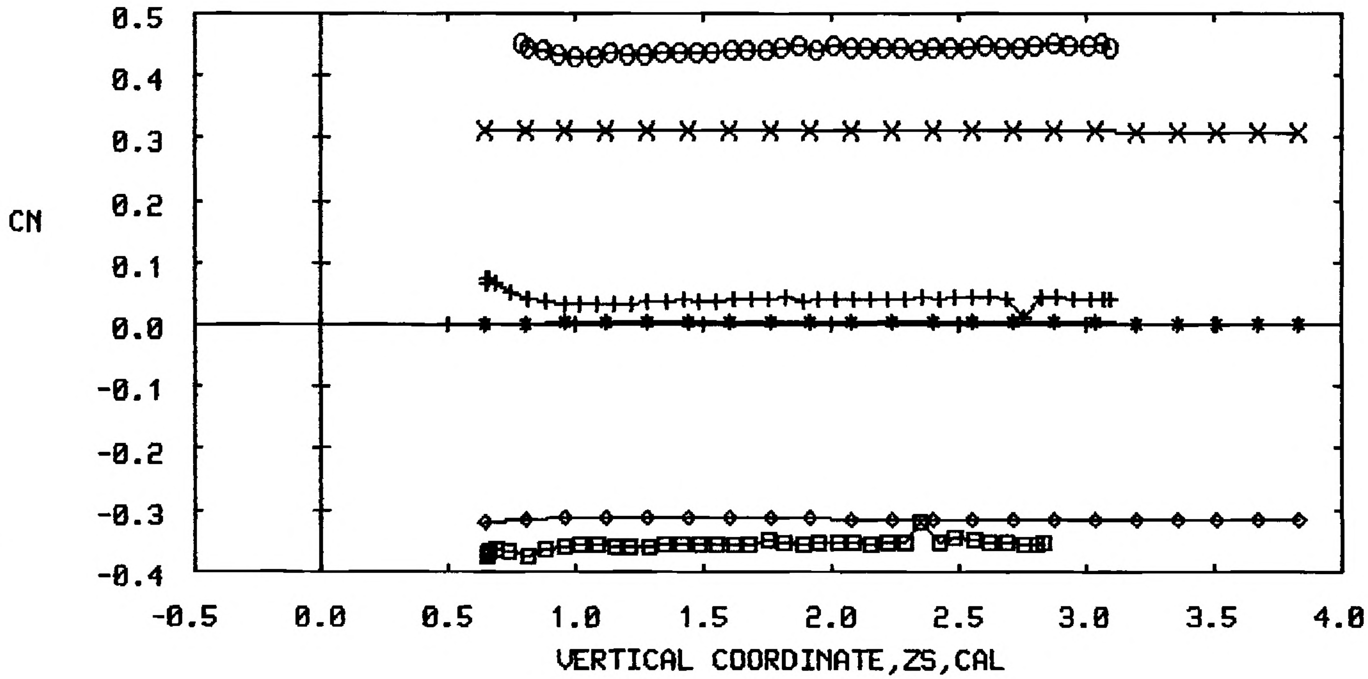

Figure 20. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1054	N2D1	5.00	0.80	3.95	0.00	0.10
*	I1054	N2D1	5.00	0.80	3.95	0.00	0.00
O	A2054	N2D1	5.00	0.80	3.95	9.98	0.10
X	I2054	N2D1	5.00	0.80	3.95	10.00	0.00
□	A1055	N2D1	5.00	0.80	3.95	-10.00	0.10
◇	I1055	N2D1	5.00	0.80	3.95	-10.00	0.00

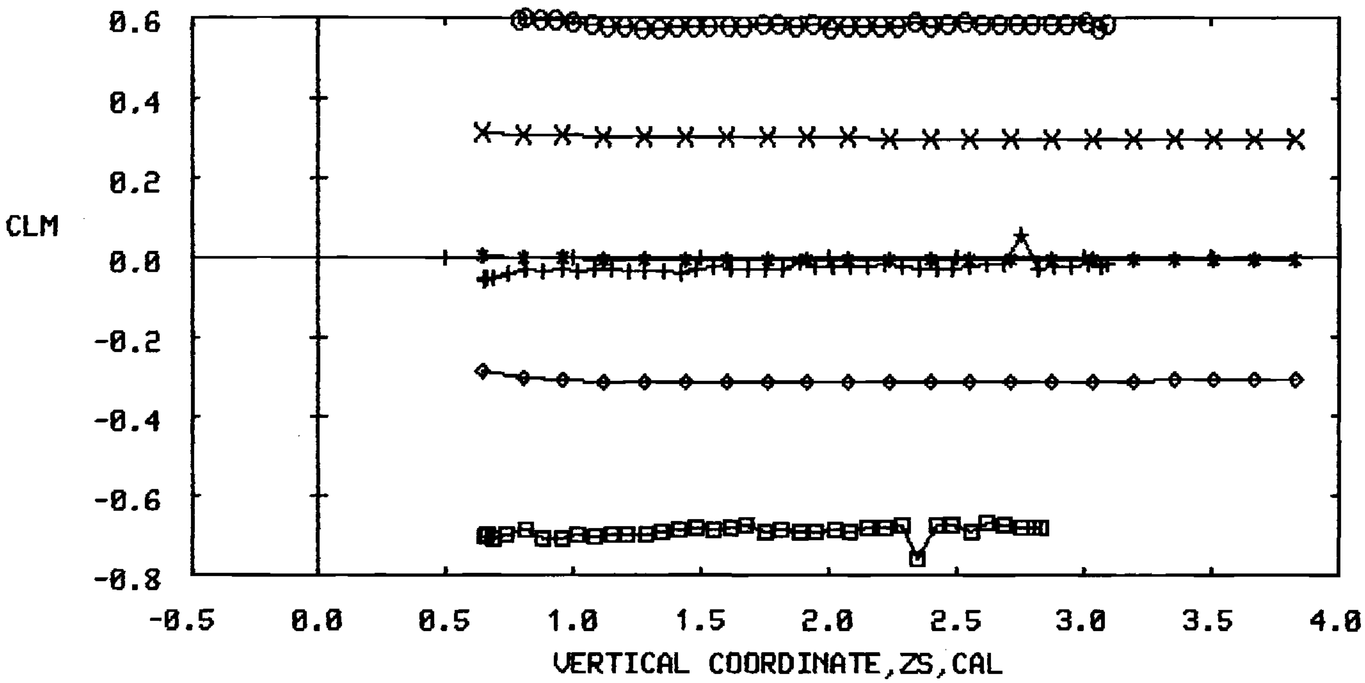

Figure 21. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

at the large angles of attack whereas their inclusion in the blunt body calculations would have degraded the agreement.

Experimental data and analytical results for the sharp nose submissile separating vertically from the front submissile bay are presented in Figures 22 and 23. Submissile separation at this location is strongly affected by the flow over the ogive nose which has larger gradients than the flow field over the cylindrical section. This is manifested in the larger gradients of the coefficients than those evident in the cases where the submissile separated from the mid cavity. Again, the source/sink distribution underpredicts the gradients when the submissile is near the dispenser missile. Also, the slender body theory does a relatively poor job of predicting the angle of attack effects on the submissile pitching moment coefficients. Dispenser missile angle of attack effects are presented in Figures 20 and 23. These figures, when compared to Figures 20 and 21, illustrate the effect of the dispenser missile angle of attack. The agreement between the analytical results and experimental data is very similar for the two sets of data. Therefore, the subsonic program also does a fair job of representing dispenser missile angle of attack effects even though it does use an empirical technique to calculate this effect.

The effect of the dispenser missile open submissile bays will be illustrated with two sets of experimental data which are shown in Figures 26 and 27 for the normal force and pitching moment coefficients. These two figures very dramatically illustrate the effect of the open cavity flow fields on the submissile aerodynamic characteristics. This data comparison shows that for any analytical technique to be successful, the cavity flow field must be simulated in some manner. The source/sink distributions which are used to represent this dispenser missile volume effect cannot represent any type of vortex flow which occurs inside the cavity. However,

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1056	N2D1	5.00	0.80	2.38	-0.01	0.10
*	I1056	N2D1	5.00	0.80	2.38	0.00	0.00
O	A2056	N2D1	5.00	0.80	2.38	9.98	0.10
X	I2056	N2D1	5.00	0.80	2.38	10.00	0.00
□	A1057	N2D1	5.00	0.80	2.38	-9.99	0.10
◇	I1057	N2D1	5.00	0.80	2.38	-10.00	0.00

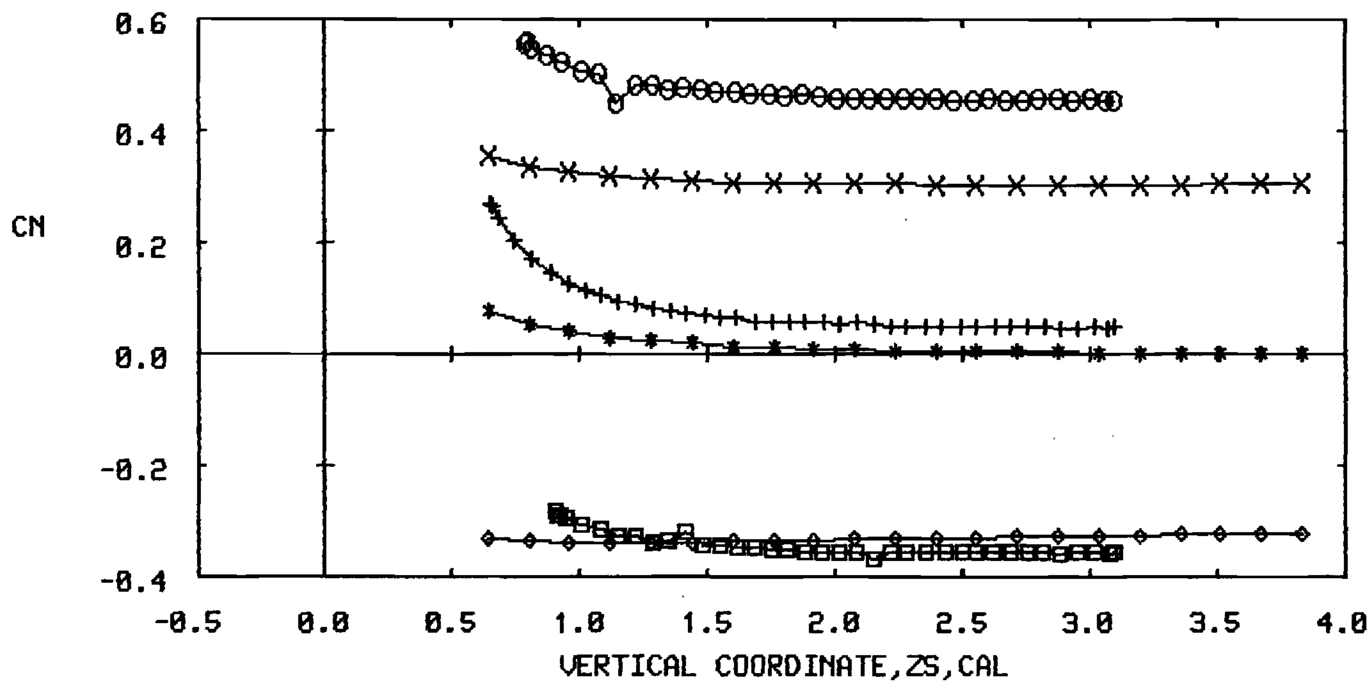

Figure 22. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1056	N2D1	5.00	0.80	2.38	-0.01	0.10
*	I1056	N2D1	5.00	0.80	2.38	0.00	0.00
O	A2056	N2D1	5.00	0.80	2.38	9.98	0.10
X	I2056	N2D1	5.00	0.80	2.38	10.00	0.00
□	A1057	N2D1	5.00	0.80	2.38	-9.99	0.10
◇	I1057	N2D1	5.00	0.80	2.38	-10.00	0.00

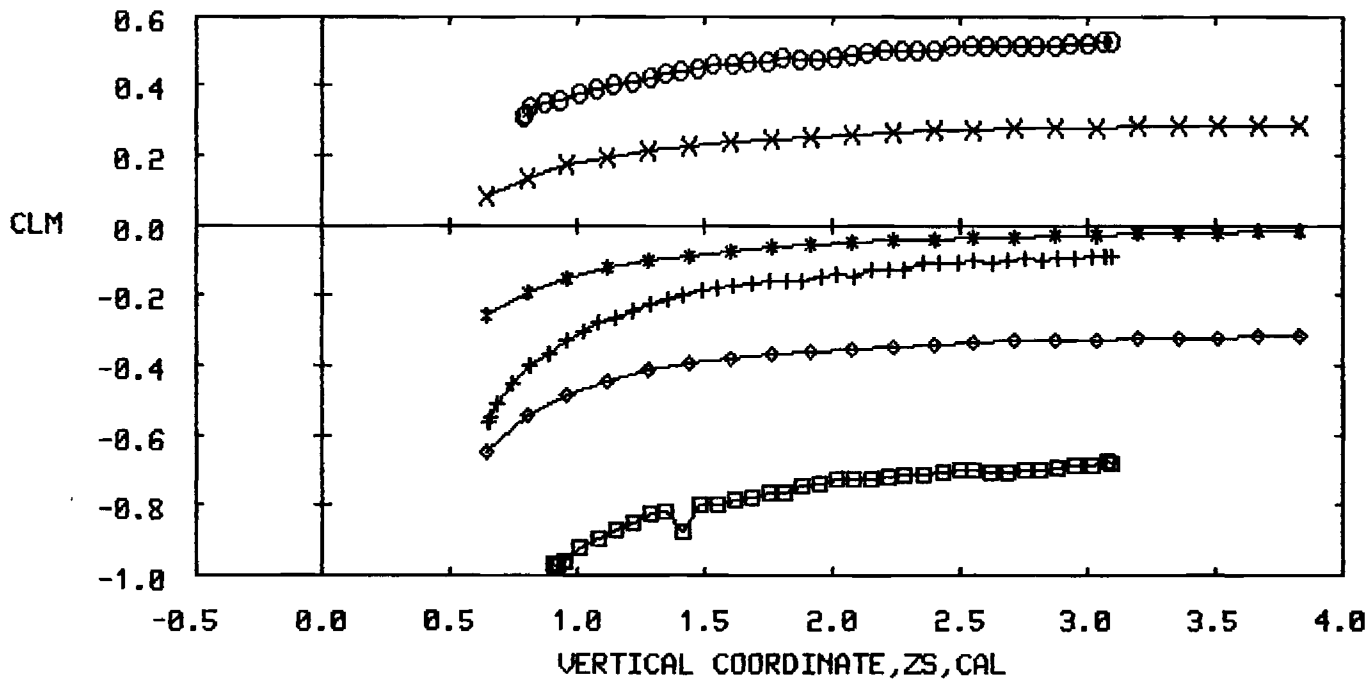

Figure 23. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1088	N2D1	5.00	0.80	3.95	0.00	5.04
*	I1088	N2D1	5.00	0.80	3.95	5.00	5.00
O	A2088	N2D1	5.00	0.80	3.95	10.01	5.04
X	I2088	N2D1	5.00	0.80	3.95	15.00	5.00
□	A3088	N2D1	5.00	0.80	3.95	-9.97	5.04
◇	I3088	N2D1	5.00	0.80	3.95	-5.00	5.00

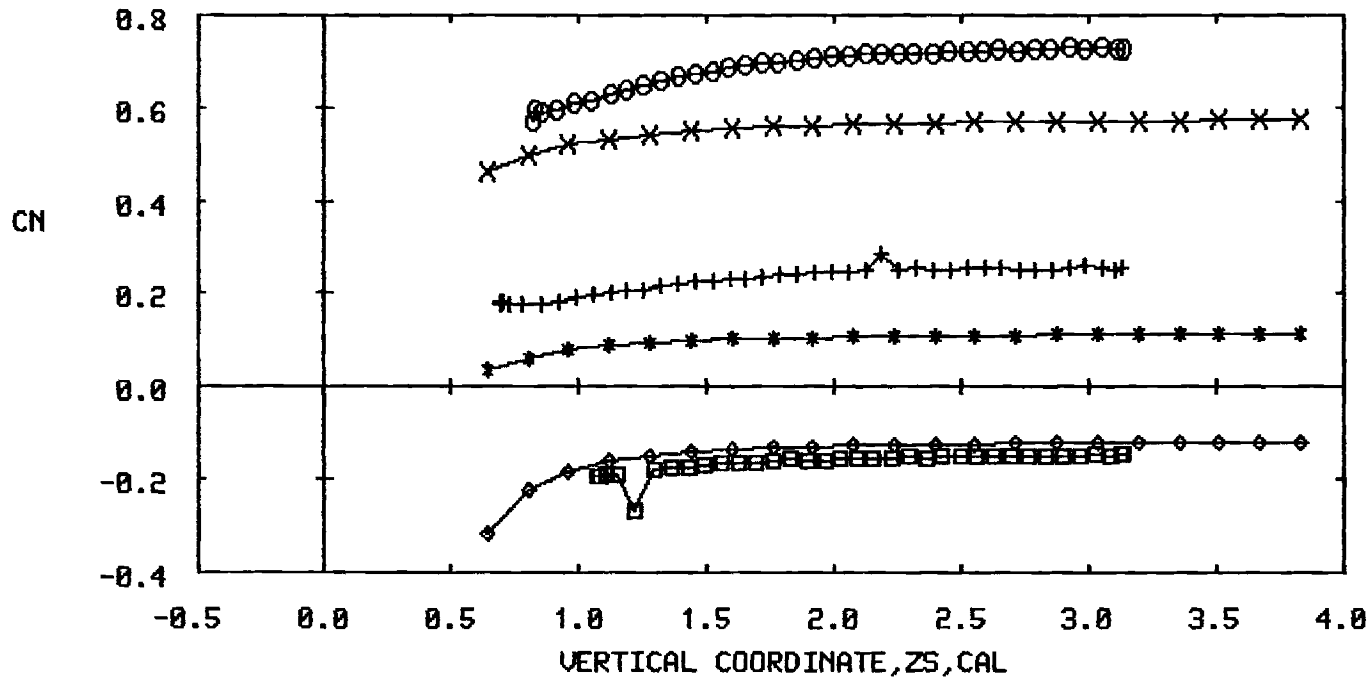

Figure 24. Comparison of normal force coefficients for the submissile, body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

SYM	RUN	CONFIGURATION	MODE	MACH	XS	ALPS	ALPD
+	A1088	N2D1	5.00	0.80	3.95	0.00	5.04
*	I1088	N2D1	5.00	0.80	3.95	5.00	5.00
O	A2088	N2D1	5.00	0.80	3.95	10.01	5.04
X	I2088	N2D1	5.00	0.80	3.95	15.00	5.00
□	A3088	N2D1	5.00	0.80	3.95	-9.97	5.04
◇	I3088	N2D1	5.00	0.80	3.95	-5.00	5.00

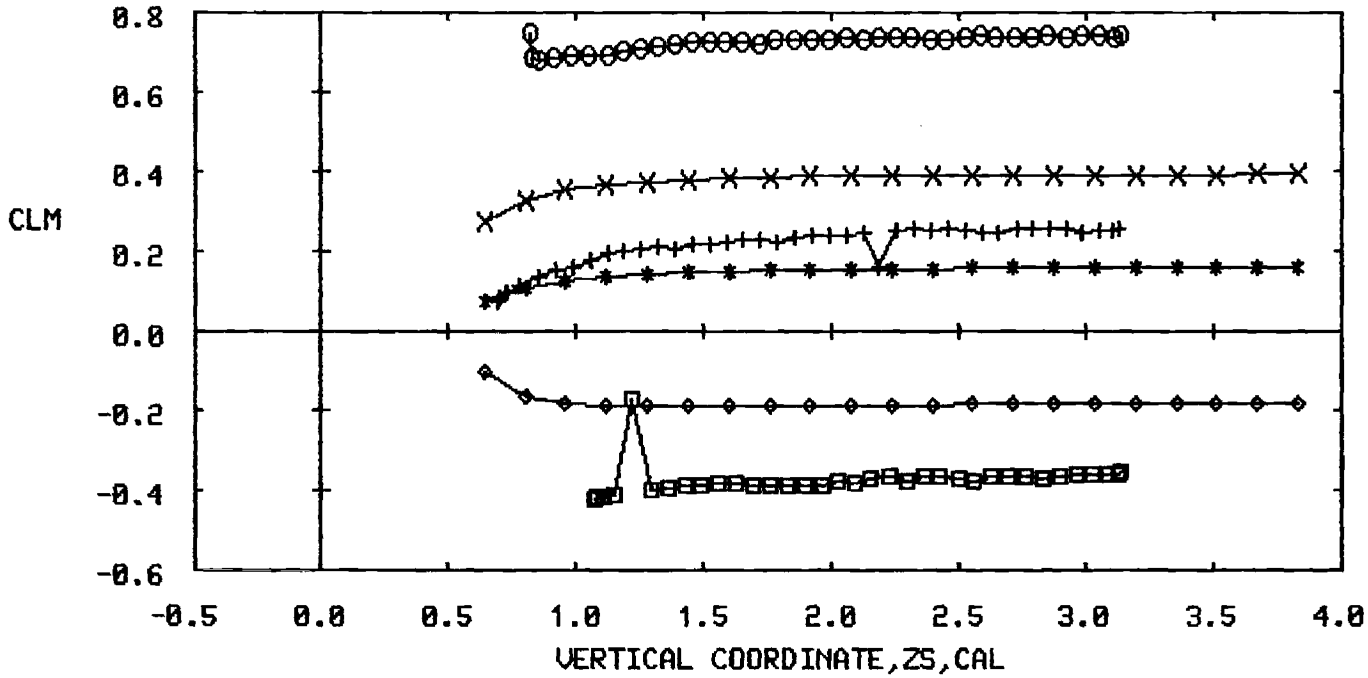

Figure 25. Comparison of pitching moment coefficients for the submissile body alone, N2, in the presence of the dispenser missile, D1, with the relative vertical location varying.

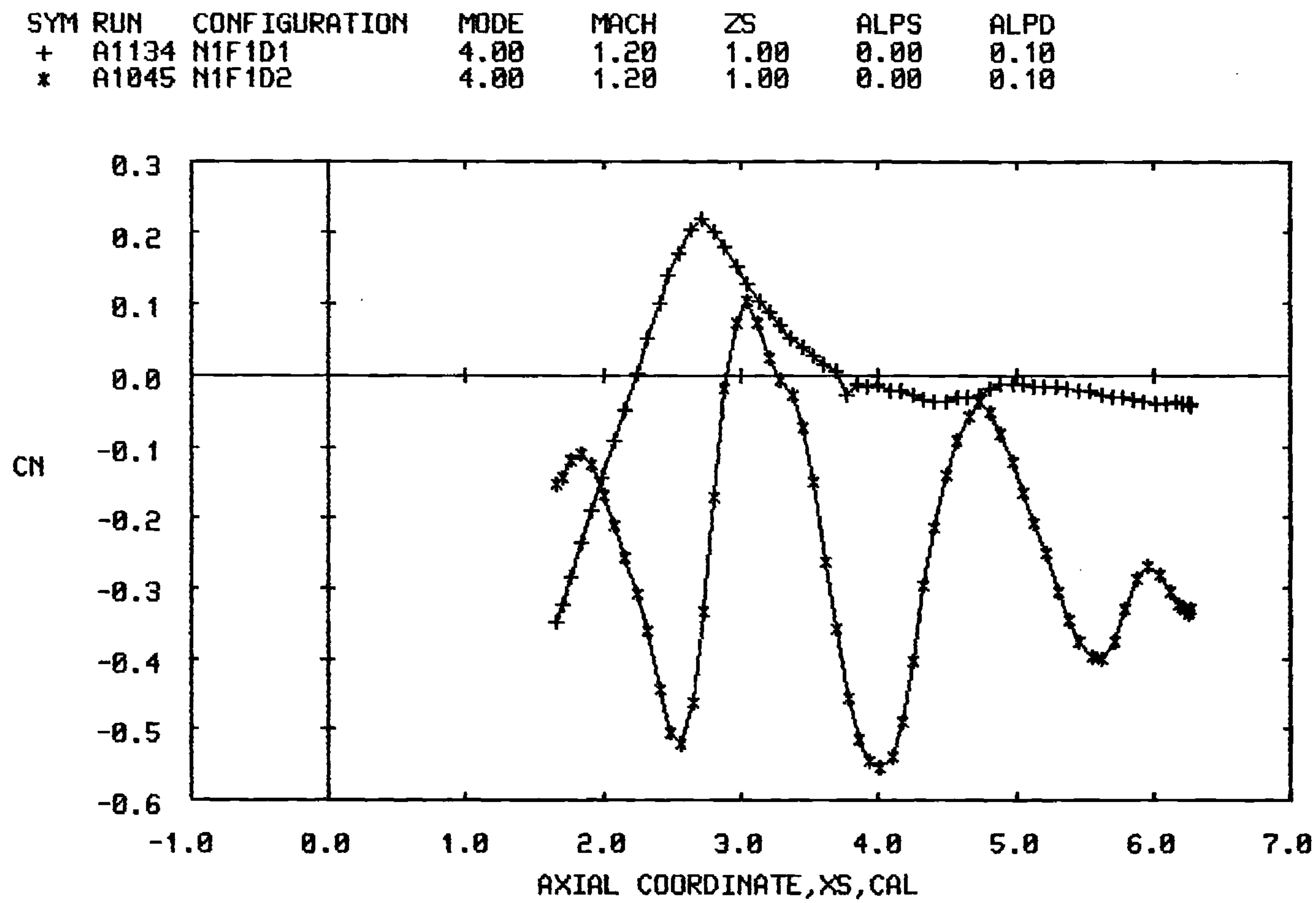

Figure 26. Comparison of normal force coefficients vs. axial location for the submissile, body plus fins, N1F1, for dispenser configurations D1 and D2.

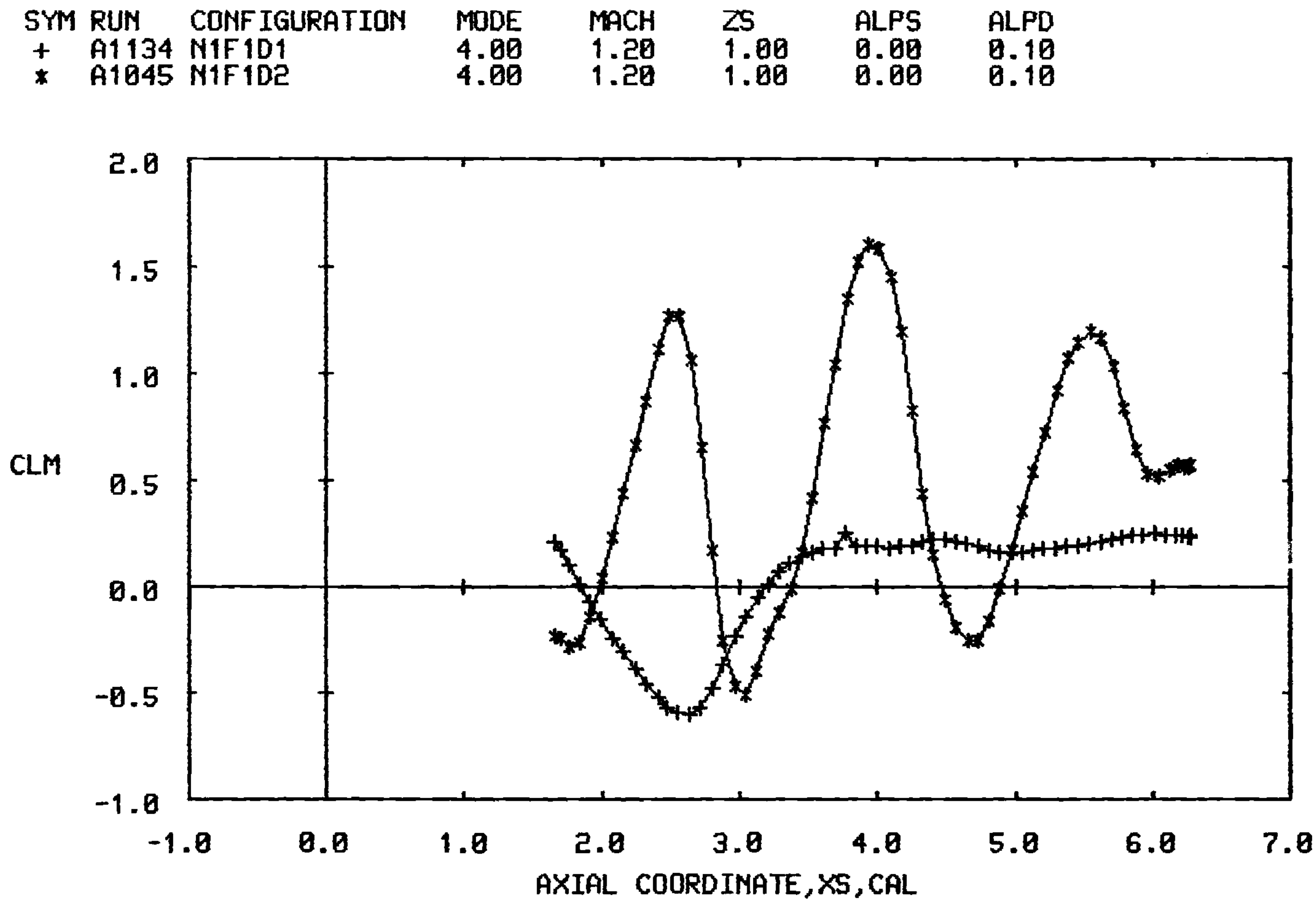

Figure 27. Comparison of pitching moment coefficients vs. axial location for the submissile, body plus fins, N1F1, for dispenser configurations D1 and D2.

this type of solution can be used to represent a mean streamline shape which essentially bounds the cavity flow field. Such bounding streamlines could be developed empirically using the data available in the 'SUBMIS' data base or they could be obtained experimentally from a specially designed test program.

8.0 SUMMARY AND CONCLUSIONS

Computer programs developed by Nielsen Engineering and Research, Inc. for store separation from aircraft have been modified to adapt them to the Perkin-Elmer 3230 and to optimize their operation for submissiles separating from dispenser missiles. Program sizes have been minimized by deletion of all program elements not required by Missile Command problems. Input data calculations have been automated wherever possible and the input format has been completely revised to be "user friendly" on the Perkin-Elmer. An option to write binary output files in the 'SUBMIS' data base format has been included in the programs and these files are compatible with the various data base programs. Calculations have been performed for a significant number of cases, exercising the various operational options, for both the supersonic and subsonic programs. A number of these calculations are presented herein and compared with data available in the 'SUBMIS' data base. Based upon these data comparisons, presented in Figures 5 through 27, the following conclusions have been reached:

A. Supersonic Program

1. The source/sink representation of the dispenser missile fuselage volume effects does a good job of calculating the interference effect on the submissile aerodynamic data.

2. The doublet representation of the dispenser missile angle of attack calculates this perturbation interference effect very well.

3. Slender body calculations for the submissile aerodynamic characteristics at large angles of attack show somewhat random quantitative agreement with experimental data.

4. Mutual interference which occurs when the dispenser and submissile are in close proximity is not considered by the program and it is, obviously, a significant factor in some of the experimental data.

5. Experimental data for the open bay configuration show a large open/closed bay effect. Therefore, for the program to be effective on the open bay configuration, an open bay flow simulation model must be developed.

B. Subsonic Program

6. The source/sink representation of the dispenser missile volume effects consistently underpredicts the experimental submissile interference effect. The underprediction is more pronounced for the more forward submissile locations

7. The empirical treatment of the dispenser missile angle of attack does not represent this effect as well as the doublet solution of the supersonic program.

8. Slender body calculations for the submissile aerodynamic characteristics at large angles of attack show somewhat random quantitative agreement with experimental data.

9. Open bay cavity simulation models must be developed for the program to effectively calculate open bay configuration interference effects on submissile aerodynamics.

REFERENCES

1. Sims, J. L., "Submissile Simulation Development," Final Technical Report, GT/EES Project A-3439, Contract No. DAAH01-83-D-A013, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, Georgia, September 1983.

2. Lundy, T. E., Braddock, W. F., and Utreja, L. R., "Submissile Aerodynamics Study - Final Report," LMSC-HREC TR D698338, Lockheed Missiles and Space Company, Inc., Huntsville, Alabama, September 1980.

3. Dillenius, Marnix F. E., Goodwin, Frederick, K., and Nielsen, Jack N., "Prediction of Supersonic Store Separation Characteristics - Volume 1 - Theoretical Methods and Comparisons with Experiments," AFFDL-TR-76-41, Volume I, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, May 1976.

4. Goodwin, Fredrick K., Keirstead, Mary M., and Dillenius, Marnix F. E., "Prediction of Supersonic Store Separation Characteristics - Volume II - User's Manual for the Computer Program," AFFDL-TR-76-41, Volume II, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, May 1976.

5. Goodwin, Frederick K., Dillenius, Marnix F. E., and Nielsen, Jack N., "Prediction of Six-Degree-of-Freedom Store Separation Trajectories at Speeds Up to the Critical Speed - Volume I - Theoretical Methods and Comparisons with Experiments," AFFDL-TR-72-83, Volume I, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, October 1972.

6. Goodwin, Frederick K., Dillenius, Marnix F. E., "Prediction of Six-Degree-of-Freedom Store Separation Trajectories at Speeds Up to the Critical Speed - Volume II - User's Manual for the Computer Program," AFFDL-TR-72-83, Volume II, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, October 1974.

7. Dedrick, L. K., and Singelton, R., "Interim Technical Report - User's Guide for Submissile (SUBMIS) Aerodynamics Data Base," TR1044, New Technology, Inc., Huntsville, Alabama, June 1981.

8. Washington, William D., "Computer Program for Estimating Stability Derivatives of Missile Configurations," Technical Report RD-76-25, U.S. Army Missile Command, Redstone Arsenal, Alabama, May 1976.

9. Royal Aeronautical Society, "Wings, Aerodynamic Data Sheets," Volume II, S.01.03.03, S.01.03.05 and S.08.01.02, May 1954.

10. Pitts, William C., Nielsen, Jack N., and Kaattari, George E., "Lift and Center of Pressure of Wing-Body-Tail Combinations at Subsonic, Transonic, and Supersonic Speeds," National Advisory Committee for Aeronautics, Report 1307, 1957.

REFERENCES (cont.)

11. Washington, William D., "Correlation of Viscous Effects and Comparison between Experimental and Theoretical Distribution of Potential Normal Force and Pitching Moment for Bodies of Revolution at Supersonic Speeds," U.S. Army Missile Command, Report No. RD-TR-12-67, December 1967.

12. Perkins, E. W., and Jorgenson, L. H., "Comparison of Experimental and Theoretical Normal Force Distributions (Including Reynolds Number Effects) on an Ogive-Cylinder Body at Mach Number 1.98," NACA TM 3716, May 1956.

13. Perkins, E. W. and Allen, H. T., "Characteristics of Flow Over Inclined Bodies of Revolution," NACA RM A50L07, March 1951.

14. Kelly, H. R., "The Estimation of Normal Force and Pitching Moment Coefficients for Blunt-Based Bodies of Revolution at Large Angles of Attack," NOTS TM 998, May 1953.

15. Goldstein, S., Modern Developments in Fluid Dynamics, Oxford, The Clarendon Press, Volume II, 1938.

GLOSSARY OF TERMS

Symbol	Nomenclature
a_{max}	Maximum store body radius
A	Fin aspect ratio
$C_1 - C_7$	Polynomial coefficients used in Equation (1)
C_{dc}	Section drag coefficient of a circular cylinder normal to airstream
C_{dc_B}	Cross flow drag for a finite length body - Equation (13)
c_r	Fin chord at fin-body juncture
C_A	Axial force coefficient, Axial force/$q_{\infty s} S_R$ R
C	Rolling moment coefficient, rolling moment/$q_{\infty s} S_R$ R
$C_{L\alpha}$	Lift coefficient curve slope
$C_{L\alpha_F}$	Fin alone lift coefficient curve slope
C_M	Pitching moment coefficient, pitching moment/$q_{\infty s} S_R$ R
C_n	Yawing moment coefficient, yawing moment/$q_{\infty} S_R \ell_R$
C_N	Normal force coefficient, normal force/$q_{\infty s} S_R$
C_Y	Side force coefficient, side force/$q_{\infty s} S_R$
d	Body diameter
K	Ratio of lift component to lift of tail alone, for variable angle of attack
ℓ	Body length
ℓ_R	Reference length, store maximum diameter, d
ℓ_s	Submissile body length
L	Lift
M_∞	Free stream Mach number
m	Cotangent of fin leading edge sweep angle
q	Free stream dynamic pressure, $1/2 \rho_\infty V^2$

GLOSSARY OF TERMS (cont.)

Symbol	Nomenclature
r	General local body radius
r_B	Body radius at fin location
s	Semispan of fin in combination with body
S_F	Fin alone area
S_R	Reference area taken equal to body frontal area, πa^2_{max}
x	General local body point longitudinal location
α	Angle of attack
β	$\sqrt{M^2 - 1}$ for $M > 1$ and $\sqrt{1 - M^2}$ for $M < 1$
Λ	Fin sweep back angle
ρ_∞	Free stream mass density
λ	Fin taper ratio, $(C_t/C_r) \equiv 1$ for current program capability

Subscripts

D	Dispenser missile
B(F)	Body in the presence of the fin
F	Last point on a conical frustum
F(B)	Fin in the presence of the body
N	Nose
s	Submissile
SB	Slender body value
o	Denotes circular arc constants
I	First point on a conical frustum
CYL	Cylindrical segment
LT	Linear theory value

APPENDIX A

SUBSONIC SOURCE COMPUTER PROGRAM DEFINITIONS AND LISTINGS

A.1 SUBSONIC SOURCE PROGRAM INPUT DATA DEFINITIONS

GENERAL: All numerical data are unformatted, list-directed. The lines of alphanumeric text describing the configuration have the format (20A4).

Geometry data may be input in any desired system, inches, feet, calibers, or other. However, all data dependent on size or location must be entered in the same system.

Guidelines for input data which are discretionary but which may significantly affect the results are discussed in the text. These guidelines are based on experience with running the program to date.

SPECIFIC:

ITEM # 1 NCARDS (an integer)

DEF: NCARDS is the number of lines of alphanumeric text which the user requires to define the configuration. To cycle through the program, NCARDS must be 1 or greater.

NOTE: The program returns to this read after each solution is completed; another problem may be entered or the program is stopped by setting NCARDS = -1.

ITEM # 2 Q(K), K = 1, 20 [format 20A4] alphanumeric

DEF: Q(K) is NCARDS lines of text which the user supplies to define the problem, configuration, or other identification. There must be at least one line of text. May be as many more as user desires.

ITEM # 3 NSECT, FMACH (integer, real)

DEF: NSECT is the number of sections or segments required to define the total body. Each section is defined by a polynomial equation. For a blunt based body, the subsonic wake must be simulated by an additional section(s) which is closed.

FMACH is the free stream mach number for which the solution is required.

ITEM # 4 AL (real)

DEF: AL is the body length which is used to non-dimensionalize all geometric quantities. Must be actual body length. Can be any desired units; all succeeding geometric data must be same units.

ITEM # 5 XEND(K), K = 1, NSECT (real; all on one line)

DEF: XEND(K) are the x (axial) location of the downstream end of each of the sections used to describe the body. There must be NSECT values and they must be in order from nose to tail. Same units as AL.

ITEMS # 6 and # 7 are read in a DO loop and must be input for each of the NSECT sections used to describe the body. Each item is one line and, thus, there are NSECT pairs of these items.

ITEM # 6 NTYPE (integer)

DEF: NTYPE is an integer which denotes the type of body section and controls the geometric input data requirements. At present, there is the capability to handle only three types of section.

NTYPE = 1, OGIVE (circular arc) section
NTYPE = 2, Conical nose or frustum section
NTYPE = 3, Cylindrical section

ITEM # 7 Section geometric data which depends upon the type of section.

If NTYPE = 1 (circular arc)
XO, RO, CRO (real)

DEF: XO is the axial location of the center of the circular arc.
RO is the radial location of the center of the circular arc.
CRO is the radius of the circular arc.

If NTYPE = 2 (conical nose or frustum section)
XI, RI, XF, RF (real)

DEF: XI is the axial location of the upstream (initial) end of the cone or frustum.
RI is the radius of the body at XI.
XF is the axial location of the downstream (final) end of the cone or frustum.
RF is the radius of the body at XF.

If NTYPE = 3 (cylindrical section)
RCYL (REAL)

DEF: RCYL is the radius of the body at the cylindrical section.

All defined variables must be in same units as AL. Coordinate system origin at body nose tip.

ITEM # 8 XSFST, XSLST, XRMAX, XINIT, XFINAL, DELX, RMAX (all real)

DEF: XSFST is the location at which the first source will be

ITEM # 8 (cont.)

placed (see Section 2.2).

XSLST is the location at which the last source will be placed (see Section 2.4).

XRMAX is the location at which the maximum body radius is first reached.

XINIT is the initial point at which the body radius is to be calculated from the source distribution.

XFINAL is the final point at which the body radius is to be calculated from the source distribution.

DELX is the increment between points at which the body radius is to be calculated from the source distribution.

RMAX is the maximum radius of actual body. All variables in same units as AL.

ITEM # 9 NRAT (integer)

DEF: NRAT is an index which specifies the number of sections the body is divided into for specifying the source spacing along the body. The maximum value of NRAT is 5.

ITEM # 10 REND(K), K = 1, NRAT (real)

DEF: REND(K) are the x (axial) locations of the downstream end of body segments where the source spacing criteria, PERCR, is to be changed. Note that REND(K) must be NRAT values of the NSECT values of XEND(K) in ITEM # 5; i.e., source spacing can only be changed at body sections.

ITEM # 11 PERCR(K), K = 1, NRAT (real)

DEF: PERCR(K) are the values of the source spacing parameter which are to be used along the body length. This parameter is the fraction of the local body radius between sources (see Section 2.3).

APPENDIX A.2

SUBSONIC SOURCE PROGRAM INPUT DATA FILES

Dispenser Missile Input Data File; DISPENS1.INP

```
6
THIS INPUT DATA FILE IS FOR THE ARMY MISSILE COMMAND DISPENSER
MISSILE.  THE ACTUAL CONFIGURATION IS AN OGIVE-CYLINDER WHICH IS
35.62 INCHES LONG.  FOR THE SUBSONIC SOURCE DISTRIBUTION THE WAKE IS
REPRESENTED BY AN OGIVE WITH DIMENSIONS EQUAL TO THOSE OF THE
NOSE OGIVE.  ALL LOCATIONS AND DIMENSIONS IN THIS FILE ARE IN INCHES
AND IT IS TO BE USED WITH THE SUBSRCE PROGRAM.
3,0.8
35.62
11.25,35.62,46.87
1
11.25,-32.808,34.688
3
1.880
1
35.62,-32.808,34.688
0.021372,46.6622,11.25,0.3562,46.306,0.7124,1.880
2
11.25,46.87
0.7,1.0
-1
```

Submissile Input Data File; SUBMISA1.INP

```
6
THIS INPUT FILE IS FOR THE TWO-CALIBER OGIVE NOSE SUBMISSILE
CONFIGURATION OF THE "ASUBMIS" DATA BASE.  THE FUSELAGE IS
5.593 INCHES LONG AND A 1.864 INCH LONG OGIVE HAS BEEN ADDED
TO THE BASE TO SIMULATE THE SUBSONIC WAKE. ALL LOCATIONS AND
DIMENSIONS ARE IN INCHES AND THIS INPUT FILE IS FOR THE
        "SUBSRCE" VERSION OF THE PROGRAM.
3,0.8
5.593
1.864,5.593,7.457
1
1.864,-3.495,3.961
3
0.4660
1
5.593,-3.495,3.961
 .003356,7.41073,1.8640,0.05593,7.2709,0.111860,0.466
2
1.864,7.457
0.7,1.0
-1
```

Submissile Input Data File; SUBMISB1.INP

```
6
THIS INPUT FILE IS FOR THE ONE-HALF CALIBER(HEMISPHERICAL) OGIVE
NOSE SUBMISSILE OF THE "ASUBMIS" DATA BASE. THE FUSELAGE IS
5.593 INCHES LONG AND A 1.664 INCH LONG OGIVE HAS BEEN ADDED
TO THE BASE TO SIMULATE THE SUBSONIC WAKE.  ALL LOCATIONS AND
DIMENSIONS ARE IN INCHES AND THIS INPUT FILE IS FOR THE "SUBSRCE"
             VERSION OF THE PROGRAM.
3,0.8
5.593
0.4660,5.5930,7.4570
1
0.4660,0.0,0.4660
3
0.4660
1
5.593,-3.495,3.961
0.018177,7.410725,0.4660,0.027965,7.187005,0.111860,0.4660
2
0.4660,7.4570
0.4,0.8
-1
```

Submissile Input Data File; SUBMISC1.INP

```
7
THIS INPUT FILE IS FOR THE LARGE SUBMISSILE USED IN TEST "B".
IT HAS A ONE-HALF CALIBER OGIVE(HEMISPHERICAL) NOSE AN A CYLINDRICAL
AFTERBODY.  AN OGIVE 3.750 INCHES LONG WITH A RADIUS OF CURVATURE
OF 8.0 INCHES HAS BEEN ADDED TO THE BASE OF THE SUBMISSILE TO
SIMULATE THE SUBSONIC WAKE.  ALL LOCATIONS AND DIMENSIONS ARE IN
INCHES AND THIS INPUT FILE IS FOR THE "SUBSRCE" VERSION OF THE
                         PROGRAM.
3,0.8
11.200
0.9320,11.200,14.94744927
1
0.932,0.0,0.9320
3
0.9320
1
11.200,-7.0680,8.000
0.036400,14.8400,0.9320,0.11200,14.6720,0.22400,0.932
2
0.9320,14.947744927
0.4,0.8
-1
```

Submissile Input Data File; SUBMISD1.INP

```
7
THIS INPUT DATA FILE IS FOR THE HIGH FINENESS RATIO SUBMISSILE USED
IN TEST "B".  IT HAS A ONE-HALF CALIBER OGIVE(HEMISPHERICAL) NOSE AND
A CYLINDRICAL AFTERBODY AND AN OVERALL FINENESS RATIO OF 14.  A TAIL
OGIVE WITH A RADIUS OF CURVATURE OF 3.5 INCHES WITH A LENGTH OF
1.624808 INCHES HAS BEEN ADDED TO SIMULATE THE SUBSONIC WAKE. ALL
LOCATIONS AND DIMENSIONS ARE IN INCHES AND THIS INPUT FILE IS FOR
      THE "SUBSRCE" VERSION OF THE PROGRAM.
3,0.8
11.200
0.400,11.200,12.824808
1
0.400,0.0,0.400
3
0.400
1
11.200,-3.100,3.500
0.038080,12.7120,0.400,0.0560,12.7680,0.0560,0.400
2
0.4,12.834808
0.4,0.8
-1
```

Submissile Input Data File; SUBMISE1.INP

```
7
THIS INPUT FILE IS FOR THE SMALL SUBMISSILE USED IN TEST "B".
IT HAS A ONE-HALF CALIBER OGIVE(HEMISPHERICAL) NOSE AND A CYLINDRICAL
AFTERBODY WITH AN OVERALL FINENESS RATIO OF 6.0 AND A CYLINDER
DIAMETER OF 0.6 INCHES.  AN OGIVE WITH A RADIUS OF CURVATURE OF 2.750
INCHES AND A LENGTH OF 1.2490 INCHES HAS BEEN ADDED TO THE BASE TO
SIMULATE THE SUBSONIC WAKE.  ALL LOCATIONS AND DIMENSIONS ARE IN INCHES
AND THIS INPUT FILE IS FO THE "SUBSRCE" VERSION OF THE PROGRAM.
3,0.8
3.6
0.3,3.6,4.849
1
0.300,0.0,0.300
3
0.300
1
3.600,-2.450,2.750
0.022572,4.7700,0.300,0.0360,4.7880,0.0720,0.300
2
0.3,4.849
0.3,0.8
-1
```

APPENDIX A.3

Subsonic Source Program Output File; NEARSOR.OPT

1

THIS INPUT DATA FILE IS FOR THE ARMY MISSILE COMMAND DISPENSER MISSILE. THE ACTUAL CONFIGURATION IS AN OGIVE-CYLINDER WHICH IS 35.62 INCHES LONG. FOR THE SUBSONIC SOURCE DISTRIBUTION THE WAKE IS REPRESENTED BY AN OGIVE WITH DIMENSIONS EQUAL TO THOSE OF THE NOSE OGIVE. ALL LOCATIONS AND DIMENSIONS IN THIS FILE ARE IN INCHES AND IT IS TO BE USED WITH THE SUBSRCE PROGRAM.

X/L OF END POINT OF EACH SECTION OF BODY

SECTION	1	2	3
X/L	0.31583	1.00000	1.31583

COEFFICIENTS OF POLYNOMIAL DESCRIBING EACH SECTION

SECTION	C1	C2	C3	C4	C5	C6	C7
1	-0.92106	-1.00000	0.63167	0.84860	0.00000	0.00000	1.00000
2	0.05278	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	-0.92106	-1.00000	2.00000	-0.05165	0.00000	0.00000	1.00000

FIRST SOURCE AT X/L= 0.00060
FROM 0.00060 TO 0.31583 SOURCE SPACING IS 0.70000 TIMES LOCAL RADIUS
FROM 0.31583 TO 1.31583 SOURCE SPACING IS 1.00000 TIMES LOCAL RADIUS

1

SOURCE LOCATIONS AND BODY RADIUS AND SURFACE SLOPE AT THESE LOCATIONS

X/L	0.00060	0.00075	0.00091	0.00110	0.00132	0.00157	0.00185
R/L	0.00035	0.00040	0.00045	0.00052	0.00059	0.00068	0.00077
DR/DX	0.34212	0.34195	0.34175	0.34152	0.34125	0.34095	0.34061
X/L	0.00218	0.00255	0.00297	0.00346	0.00401	0.00465	0.00537
R/L	0.00088	0.00101	0.00116	0.00132	0.00151	0.00172	0.00196
DR/DX	0.34021	0.33976	0.33925	0.33866	0.33799	0.33723	0.33636
X/L	0.00619	0.00714	0.00821	0.00943	0.01083	0.01242	0.01423
R/L	0.00224	0.00256	0.00291	0.00332	0.00378	0.00431	0.00490
DR/DX	0.33536	0.33423	0.33294	0.33147	0.32979	0.32789	0.32573
X/L	0.01628	0.01862	0.02127	0.02428	0.02768	0.03154	0.03589
R/L	0.00556	0.00632	0.00716	0.00811	0.00917	0.01036	0.01167
DR/DX	0.32327	0.32049	0.31734	0.31378	0.30976	0.30523	0.30014
X/L	0.04079	0.04631	0.05249	0.05942	0.06716	0.07577	0.08531
R/L	0.01313	0.01474	0.01650	0.01842	0.02049	0.02272	0.02510
DR/DX	0.29442	0.28802	0.28088	0.27293	0.26411	0.25437	0.24364
X/L	0.09585	0.10745	0.12014	0.13397	0.14894	0.16505	0.18229
R/L	0.02761	0.03022	0.03291	0.03565	0.03837	0.04104	0.04358
DR/DX	0.23188	0.21906	0.20513	0.19010	0.17395	0.15672	0.13844
X/L	0.20059	0.21988	0.24006	0.26099	0.28251	0.30443	0.32657
R/L	0.04594	0.04804	0.04983	0.05123	0.05221	0.05271	0.05278
DR/DX	0.11918	0.09901	0.07805	0.05641	0.03424	0.01171	0.00000
X/L	0.35824	0.38991	0.42158	0.45324	0.48491	0.51658	0.54825
R/L	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278

DR/DX	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
X/L	0.57991	0.61158	0.64325	0.67492	0.70658	0.73825	0.76992
R/L	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278
DR/DX	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
X/L	0.80159	0.83325	0.86492	0.89659	0.92826	0.95992	0.99159
R/L	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278
DR/DX	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
X/L	1.02326	1.05476	1.08550	1.11491	1.14250	1.16788	1.19080
R/L	0.05250	0.05124	0.04902	0.04598	0.04230	0.03820	0.03391
DR/DX	-0.02389	-0.05632	-0.08814	-0.11883	-0.14792	-0.17501	-0.19980
X/L	1.21114	1.22891	1.24421	1.25720	1.26812	1.27721	1.28470
R/L	0.02961	0.02549	0.02166	0.01820	0.01514	0.01249	0.01023
DR/DX	-0.22210	-0.24184	-0.25904	-0.27384	-0.28640	-0.29694	-0.30571
X/L	1.29084	1.29584	1.29990	1.30317	1.30580	1.30791	1.30960
R/L	0.00833	0.00675	0.00545	0.00439	0.00352	0.00282	0.00225
DR/DX	-0.31294	-0.31886	-0.32368	-0.32759	-0.33074	-0.33328	-0.33532

1

FOR THIS CASE THERE ARE 91 SOURCES

INCOMPRESSIBLE SOURCE DISTRIBUTION FOR MACH NUMBER 0.80

X/L	6.0000E-04	7.4547E-04	9.1185E-04	1.1021E-03	1.3197E-03	1.5684E-03
Q	6.5039E-07	-7.5160E-07	5.6776E-07	-4.0993E-07	3.0507E-07	-2.0817E-07
X/L	1.8528E-03	2.1779E-03	2.5494E-03	2.9741E-03	3.4593E-03	4.0135E-03
Q	1.6866E-07	-9.7996E-08	1.0323E-07	-3.2337E-08	7.9332E-08	1.5665E-08
X/L	4.6465E-03	5.3693E-03	6.1943E-03	7.1356E-03	8.2094E-03	9.4336E-03
Q	8.4443E-08	6.3857E-08	1.1697E-07	1.2787E-07	1.8415E-07	2.2624E-07
X/L	1.0829E-02	1.2417E-02	1.4225E-02	1.6282E-02	1.8618E-02	2.1271E-02
Q	3.0284E-07	3.8522E-07	5.0261E-07	6.4418E-07	8.3065E-07	1.0625E-06
X/L	2.4278E-02	2.7685E-02	3.1537E-02	3.5887E-02	4.0790E-02	4.6305E-02
Q	1.3583E-06	1.7265E-06	2.1868E-06	2.7535E-06	3.4483E-06	4.2877E-06
X/L	5.2495E-02	5.9424E-02	6.7159E-02	7.5766E-02	8.5310E-02	9.5853E-02
Q	5.2920E-06	6.4735E-06	7.8414E-06	9.3904E-06	1.1104E-05	1.2938E-05
X/L	1.0745E-01	1.2014E-01	1.3397E-01	1.4894E-01	1.6505E-01	1.8229E-01
Q	1.4828E-05	1.6675E-05	1.8351E-05	1.9697E-05	2.0536E-05	2.0686E-05
X/L	2.0059E-01	2.1988E-01	2.4006E-01	2.6099E-01	2.8251E-01	3.0443E-01
Q	1.9979E-05	1.8280E-05	1.5650E-05	1.1034E-05	9.8347E-06	-6.9701E-07
X/L	3.2657E-01	3.5624E-01	3.8991E-01	4.2158E-01	4.5324E-01	4.8491E-01
Q	-9.1383E-06	6.6458E-06	-7.6250E-06	7.0469E-06	-7.4151E-06	7.1677E-06
X/L	5.1658E-01	5.4825E-01	5.7991E-01	6.1158E-01	6.4325E-01	6.7492E-01
Q	-7.3384E-06	7.2205E-06	-7.2996E-06	7.2509E-06	-7.2734E-06	7.2749E-06
X/L	7.0658E-01	7.3825E-01	7.6992E-01	8.0159E-01	8.3325E-01	8.6492E-01
Q	-7.2493E-06	7.3015E-06	-7.2182E-06	7.3416E-06	-7.1633E-06	7.4220E-06
X/L	8.9659E-01	9.2826E-01	9.5992E-01	9.9159E-01	1.0233E+00	1.0548E+00

Q	-7.0353E-06	7.6483E-06	-6.6187E-06	1.0134E-05	-7.0502E-06	-1.7665E-05
X/L	1.0855E+00	1.1149E+00	1.1425E+00	1.1679E+00	1.1908E+00	1.2111E+00
Q	-2.5730E-05	-3.0863E-05	-3.2360E-05	-3.1286E-05	-2.7752E-05	-2.3590E-05
X/L	1.2289E+00	1.2442E+00	1.2572E+00	1.2681E+00	1.2772E+00	1.2847E+00
Q	-1.8549E-05	-1.4597E-05	-1.0268E-05	-8.0394E-06	-4.6989E-06	-4.4790E-06
X/L	1.2908E+00	1.2958E+00	1.2999E+00	1.3032E+00	1.3058E+00	1.3079E+00
Q	-1.2695E-06	-3.2921E-06	1.2754E-06	-4.0710E-06	4.2935E-06	-7.2796E-06
X/L	1.3096E+00					
Q	7.6152E-06					

1

SHAPE CALCULATED FROM SOURCE DISTRIBUTION AND POLYNOMIALS

X/L	0.01000	0.03000	0.05000	0.07000	0.09000	0.11000	0.13000
R/L(S.D.)	0.00412	0.00990	0.01588	0.02124	0.02629	0.03082	0.03495
R/L(POLY)	0.00351	0.00989	0.01579	0.02124	0.02623	0.03078	0.03488
X/L	0.15000	0.17000	0.19000	0.21000	0.23000	0.25000	0.27000
R/L(S.D.)	0.03861	0.04185	0.04464	0.04706	0.04902	0.05056	0.05175
R/L(POLY)	0.03856	0.04180	0.04461	0.04701	0.04899	0.05055	0.05170
X/L	0.29000	0.31000	0.33000	0.35000	0.37000	0.39000	0.41000
R/L(S.D.)	0.05252	0.05283	0.05278	0.05278	0.05283	0.05283	0.05278
R/L(POLY)	0.05244	0.05276	0.05278	0.05278	0.05278	0.05278	0.05278
X/L	0.43000	0.45000	0.47000	0.49000	0.51000	0.53000	0.55000
R/L(S.D.)	0.05283	0.05283	0.05278	0.05283	0.05283	0.05278	0.05283
R/L(POLY)	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278
X/L	0.57000	0.59000	0.61000	0.63000	0.65000	0.67000	0.69000
R/L(S.D.)	0.05283	0.05278	0.05278	0.05288	0.05278	0.05278	0.05288
R/L(POLY)	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278
X/L	0.71000	0.73000	0.75000	0.77000	0.79000	0.81000	0.83000
R/L(S.D.)	0.05278	0.05278	0.05283	0.05283	0.05278	0.05283	0.05283
R/L(POLY)	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278
X/L	0.85000	0.87000	0.89000	0.91000	0.93000	0.95000	0.97000
R/L(S.D.)	0.05278	0.05283	0.05283	0.05278	0.05283	0.05283	0.05278
R/L(POLY)	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278	0.05278
X/L	0.99000	1.01000	1.03000	1.05000	1.07000	1.09000	1.11000
R/L(S.D.)	0.05278	0.05278	0.05237	0.05154	0.05025	0.04855	0.04649
R/L(POLY)	0.05278	0.05273	0.05232	0.05149	0.05026	0.04861	0.04655
X/L	1.13000	1.15000	1.17000	1.19000	1.21000	1.23000	1.25000
R/L(S.D.)	0.04397	0.04098	0.03757	0.03371	0.02938	0.02453	0.01917
R/L(POLY)	0.04406	0.04116	0.03783	0.03406	0.02987	0.02523	0.02014
X/L	1.27001	1.29001					
R/L(S.D.)	0.01299	0.00536					
R/L(POLY)	0.01460	0.00860					

1

APPENDIX A.4

Subsonic Source Program Source Output File;
NEARSOR .SRC

```
NSECT = 3
  X/L        0.31583   1.00000   1.31583
      1     -0.92106  -1.00000   0.63167   0.84860   0.00000   0.00000   1.00000
      2      0.05278   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
      3     -0.92106  -1.00000   2.00000  -0.05165   0.00000   0.00000   1.00000

NSURC =   91

   X/L     6.0000E-04  7.4547E-04  9.1185E-04  1.1021E-03  1.3197E-03  1.5684E-03
   Q       6.5039E-07 -7.5160E-07  5.6776E-07 -4.0993E-07  3.0507E-07 -2.0817E-07

   X/L     1.8528E-03  2.1779E-03  2.5494E-03  2.9741E-03  3.4593E-03  4.0135E-03
   Q       1.6866E-07 -9.7996E-08  1.0323E-07 -3.2337E-08  7.9332E-08  1.5665E-08

   X/L     4.6465E-03  5.3693E-03  6.1943E-03  7.1356E-03  8.2094E-03  9.4336E-03
   Q       8.4443E-08  6.3857E-08  1.1697E-07  1.2787E-07  1.8415E-07  2.2624E-07

   X/L     1.0829E-02  1.2417E-02  1.4225E-02  1.6282E-02  1.8618E-02  2.1271E-02
   Q       3.0284E-07  3.8522E-07  5.0261E-07  6.4418E-07  8.3065E-07  1.0625E-06

   X/L     2.4278E-02  2.7685E-02  3.1537E-02  3.5887E-02  4.0790E-02  4.6305E-02
   Q       1.3583E-06  1.7265E-06  2.1868E-06  2.7535E-06  3.4483E-06  4.2877E-06

   X/L     5.2495E-02  5.9424E-02  6.7159E-02  7.5766E-02  8.5310E-02  9.5853E-02
   Q       5.2920E-06  6.4735E-06  7.8414E-06  9.3904E-06  1.1104E-05  1.2938E-05

   X/L     1.0745E-01  1.2014E-01  1.3397E-01  1.4894E-01  1.6505E-01  1.8229E-01
   Q       1.4828E-05  1.6675E-05  1.8351E-05  1.9697E-05  2.0536E-05  2.0686E-05

   X/L     2.0059E-01  2.1988E-01  2.4006E-01  2.6099E-01  2.8251E-01  3.0443E-01
   Q       1.9979E-05  1.8280E-05  1.5650E-05  1.1034E-05  9.8347E-06 -6.9701E-07

   X/L     3.2657E-01  3.5824E-01  3.8991E-01  4.2158E-01  4.5324E-01  4.8491E-01
   Q      -9.1383E-06  6.6458E-06 -7.6250E-06  7.0469E-06 -7.4151E-06  7.1677E-06

   X/L     5.1658E-01  5.4825E-01  5.7991E-01  6.1158E-01  6.4325E-01  6.7492E-01
   Q      -7.3384E-06  7.2205E-06 -7.2996E-06  7.2509E-06 -7.2734E-06  7.2749E-06

   X/L     7.0658E-01  7.3825E-01  7.6992E-01  8.0159E-01  8.3325E-01  8.6492E-01
   Q      -7.2493E-06  7.3015E-06 -7.2182E-06  7.3416E-06 -7.1633E-06  7.4220E-06

   X/L     8.9659E-01  9.2826E-01  9.5992E-01  9.9159E-01  1.0233E+00  1.0548E+00
   Q      -7.0353E-06  7.6483E-06 -6.6187E-06  1.0134E-05 -7.0502E-06 -1.7665E-05

   X/L     1.0855E+00  1.1149E+00  1.1425E+00  1.1679E+00  1.1908E+00  1.2111E+00
   Q      -2.5730E-05 -3.0863E-05 -3.2360E-05 -3.1286E-05 -2.7752E-05 -2.3590E-05

   X/L     1.2289E+00  1.2442E+00  1.2572E+00  1.2681E+00  1.2772E+00  1.2847E+00
   Q      -1.8549E-05 -1.4597E-05 -1.0268E-05 -8.0394E-06 -4.6989E-06 -4.4790E-06

   X/L     1.2908E+00  1.2958E+00  1.2999E+00  1.3032E+00  1.3058E+00  1.3079E+00
   Q      -1.2695E-06 -3.2921E-06  1.2754E-06 -4.0710E-06  4.2935E-06 -7.2796E-06

   X/L     1.3096E+00
   Q       7.6152E-06
```

APPENDIX A.5

Subsonic Source Program FORTRAN Listing; NEARSOR .FTN

```
$BATCH
C     PROGRAM SOURCE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)               SORB1  1
C     LEVEL 2,C
      COMMON/CC/C(100,101)
C     AXISYMMETRIC SOURCE DISTRIBUTION PROGRAM                            SORB1  2
C                                                                         SORB1  3
      DIMENSION Q(100),RP(15),XS(100),XEND(15),COEF(15,7),
     *RA(100),XC(100),TB(100),JJJ(15),RL(100),DRDX(100),XX(15),RR(15)
      DIMENSION REND(5),PERCR(5)                                          SORB1  6
  701 FORMAT(1H1)
  704 FORMAT(1X,20A4)                                                     SORB1 10
  703 FORMAT(20A4)
  706 FORMAT(//1X,40HX/L OF END POINT OF EACH SECTION OF BODY/)           SORB1 12
  714 FORMAT(//1X,23HFOR THIS CASE THERE ARE,I4,8H SOURCES)               SORB1 13
  716 FORMAT(//1X,50HINCOMPRESSIBLE SOURCE DISTRIBUTION FOR MACH NUMBER,SORB1 14
     1F5.2)                                                               SORB1 15
  719 FORMAT(//1X,57HSHAPE CALCULATED FROM SOURCE DISTRIBUTION AND POLYNSORB1 16
     1OMIALS)                                                             SORB1 17
  722 FORMAT(/2X,7HSECTION,5X,I2,6(8X,I2))                                SORB1 18
  723 FORMAT(2X,3HX/L,5X,7F10.5)                                          SORB1 19
  724 FORMAT(//1X,51HCOEFFICIENTS OF POLYNOMIAL DESCRIBING EACH SECTION SORB1 20
     1//3X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,SORB1 21
     22HC7)                                                               SORB1 22
  725 FORMAT(5X,I2,3X,7F10.5)                                             SORB1 23
  726 FORMAT(//1X,20HFIRST SOURCE AT X/L=,F8.5)                           SORB1 24
  727 FORMAT(//1X,69HSOURCE LOCATIONS AND BODY RADIUS AND SURFACE SLOPE SORB1 25
     1AT THESE LOCATIONS)                                                 SORB1 26
  728 FORMAT(/1X,3HX/L,6X,7F10.5)                                         SORB1 27
  729 FORMAT(1X,9HR/L(S.D.),    7F10.5)                                   SORB1 28
  730 FORMAT(3X,5HDR/DX,2X,7F10.5)                                        SORB1 29
  731 FORMAT(/3X,3HX/L,2X,6(1PE12.4))                                     SORB1 30
  732 FORMAT(3X,1HQ,4X,6(1PE12.4))                                        SORB1 31
  733 FORMAT(1X,4HFROM,F10.5,2X,2HTO,F10.5,2X,17HSOURCE SPACING IS,F10.5SORB1 32
     1,2X,18HTIMES LOCAL RADIUS)                                          SORB1 33
  734 FORMAT(1X,9HR/L(POLY),    7F10.5)                                   SORB1 34
  735 FORMAT(/3X,3HX/L,4X,7F10.5)                                         SORB1 35
  736 FORMAT( 3X,3HR/L,4X,7F10.5)                                         SORB1 36
C                                                                         SORB1 37
C     READ AND PRINT INPUT DATA                                           SORB1 38
C                                                                         SORB1 39
       CALL CARCON(6,1)
    4 WRITE (6,701)                                                       SORB1 40
      READ (5, *  ) NCARDS                                                SO
      IF (NCARDS .GT. 0 ) GO TO 6
      STOP                                                                SORB1 43
    6 CONTINUE                                                            SORB1 44
      DO 1 J=1,NCARDS                                                     SORB1 45
      READ (5,703) (Q(K),K=1,20)                                          SORB1 46
    1 WRITE (6,704) (Q(K),K=1,20)                                         SORB1 47
      READ (5, * ) NSECT,FMACH                                            SORB1 48
C
C     READ THE REFERENCE LENGTH FOR THE COEFFICIENTS OF THE BODY
C     SEGMENT POLYNOMIALS - THIS MUST BE THE ACTUAL LENGTH OF THE
C     BODY.
C
      READ(5,*) AL
C
      READ (5, * ) (XEND(K),K=1,NSECT)                                    SORB1 49
      DO 5 K=1,NSECT
    5 XEND(K) = XEND(K)/AL
      WRITE (6,706)                                                       SORB1 50
      BETA=SQRT(1.0-FMACH**2)                                             SORB1 51
```

```
      NEND=0
      DO 100 K=1,15
  100 JJJ(K)=K
      WRITE (6,722) (JJJ(K),K=1,NSECT)
      WRITE (6,723) (XEND(K),K=1,NSECT)
C
      WRITE (7,801) NSECT
  801 FORMAT('NSECT =',I2)
      WRITE (7,723) (XEND(K),K=1,NSECT)
C
C     CAUTION! SUBROUTINE GEOMET, CALLED BELOW, HAS AN INPUT
C     REQUIREMENT WHICH DEPENDS UPON THE TYPE OF BODY SEGMENT.
      DO 2 J=1,NSECT
      READ (5,*) NTYPE
    2 CALL GEOMET(J,NTYPE,COEF,AL)
      WRITE (6,724)
      DO 3 J=1,NSECT
      WRITE (6,725) J,(COEF(J,K), K=1,7)
C
    3 WRITE (7,725) J,(COEF(J,K), K=1,7)
      READ (5, * ) XSFST,XSLST,XRMAX,XINIT,XFINAL,DELX,RMAX
      XSFST = XSFST/AL
      XSLST = XSLST/AL
      XRMAX = XRMAX/AL
      XINIT = XINIT/AL
      XFINAL = XFINAL/AL
      DELX  = DELX/AL
      RMAX  = RMAX/AL
      RMAX=RMAX*BETA
      READ(5, * ) NRAT
      READ(5, * ) (REND(K),K=1,NRAT)
      DO 7 K=1,NRAT
    7 REND(K) = REND(K)/AL
      READ(5, * ) (PERCR(K),K=1,NRAT)
      WRITE(6,726) XSFST
      SXM=XSFST
      DO 210 L=1,NRAT
      SX=REND(L)
      WRITE(6,733) SXM,SX,PERCR(L)
  210 SXM=SX
C
C     CALCULATE SOURCE LOCATIONS AND BODY RADIUS AND SURFACE SLOPE
C         AT THESE LOCATIONS
C
      XZ=XSFST
      J=1
   10 CALL SHAPE (XZ,NSECT,XEND,COEF,RL(J),DRDX(J))
      XS(J)=XZ
      DO 220 K=1,NRAT
      IF (XZ.GT.REND(K)) GO TO 220
      KRAT=K
      GO TO 221
  220 CONTINUE
      KRAT=NRAT
  221 XZ=XZ+PERCR(KRAT)*RL(J)*BETA
      J=J+1
      IF(J.LT.101) GO TO 20
      WRITE (6,708)
  708 FORMAT (//1X,21HMORE THAN 100 SOURCES)
      NSORC=100
      NEND=1
      GO TO 25
   20 IF (XZ.LE.XSLST) GO TO 10
```

```
      NSORC=J-1
      WRITE (6,701)
      WRITE (6,727)
   25 NA=NSORC/7
      NB=7*NA
      IF (NB.LT.NSORC) NA=NA+1
      DO 101 J=1,NA
      NB=7*(J-1)+1
      NC=NB+6
      IF (NC.GT.NSORC) NC=NSORC
      WRITE (6,735) (XS(N),N=NB,NC)
      WRITE (6,736) (RL(N),N=NB,NC)
  101 WRITE (6,730) (DRDX(N),N=NB,NC)
      IF (NEND.EQ.1) GO TO 4
C
C     CALCULATE R/L AND DR/DX AT LOCATIONS WHERE FLOW ANGLE IMPOSED
C
      NM2=NSORC-2
      NM1=NSORC-1
      NP1=NSORC+1
      J=0
   31 J=J+1
      IF (XS(J).LE.XRMAX) GO TO 31
      NPR=J-4
      DO 34 J=1,NM2
      JP1=J+1
      IF (J.GT.NPR) GO TO 33
      XZ=0.5*(XS(JP1)+XS(J))
      XC(J)=XZ
      GO TO 36
   33 JP2=J+2
      XZ=0.5*(XS(JP2)+XS(JP1))
      XC(J)=XZ
   36 CALL SHAPE(XZ,NSECT,XEND,COEF,RA(J),TB(J))
      RA(J)=RA(J)*BETA
   34 TB(J)=TB(J)*BETA
C
C     CALCULATE SOURCE DISTRIBUTION
C
C     CALCULATE COEFFICIENT MATRIX AND RIGHT HAND SIDE
C
      EXA=1.5
      EXB=2.0
      DO 40 J=1,NM2
      XZ=XC(J)
      XR=RA(J)
      SL=TB(J)
      XRS=XR*XR
      DO 40 K=1,NSORC
   40 C(J,K)=(XR-SL*(XZ-XS(K)))/(((XZ-XS(K))**2+XRS)**EXA)
      DO 41 J=1,NSORC
      C(NM1,J)=1.0
   41 C(NSORC,J)=1.0/(XS(J)**EXB)
      DO 42 J=1,NM2
   42 C(J,NP1)=TB(J)
      C(NM1,NP1)=0.0
      C(NSORC,NP1)=1.0
C
C     SOLVE FOR SOURCE STRENGTHS
C
      CALL INVERS(C,1,NSORC,100,101)
      DO 43 K=1,NSORC
   43 Q(K)=C(K,NP1)
```

```
C                                                                       SORB1159
C     OUTPUT SOURCE LOCATIONS AND STRENGTHS                             SORB1160
C                                                                       SORB1161
      WRITE (6,701)                                                     SORB1162
      WRITE (6,714) NSORC                                               SORB1163
C
      WRITE (7,802) NSORC
  802 FORMAT(/,'NSORC =',I4)
C
      WRITE (6,716) FMACH                                               SORB1164
      NAA=NSORC/6                                                       SORB1165
      NBB=6*NAA                                                         SORB1166
      IF (NBB.LT.NSORC) NAA=NAA+1                                       SORB1167
      DO 102 J=1,NAA                                                    SORB1168
      NB=6*(J-1)+1                                                      SORB1169
      NC=NB+5                                                           SORB1170
      IF (NC.GT.NSORC) NC=NSORC                                         SORB1171
      WRITE (6,731) (XS(N),N=NB,NC)                                     SORB1172
      WRITE (6,732) (Q(N),N=NB,NC)                                      SORB1173
      WRITE (7,731) (XS(N),N=NB,NC)
  102 WRITE (7,732) (Q(N),N=NB,NC)
C                                                                       SORB1174
C     CALCULATE BODY SHAPE FROM SOURCE DISTRIBUTION                     SORB1175
C                                                                       SORB1176
      X=XINIT                                                           SORB1177
      EPS=5.0E-07                                                       SORB1178
      WRITE (6,701)                                                     SORB1179
      WRITE (6,719)                                                     SORB1180
      J=0                                                               SORB1181
   55 R=RMAX                                                            SORB1182
      J=J+1                                                             SORB1183
      CALL SHAPE(X,NSECT,XEND,COEF,RP(J),DR)                            SORB1184
      M=1                                                               SORB1185
   52 PSI=R*R*0.5                                                       SORB1186
      DO 51 K=1,NSORC                                                   SORB1187
      XM=X-XS(K)                                                        SORB1188
   51 PSI=PSI-Q(K)*(1.0+XM/SQRT(XM*XM+R*R))                             SORB1189
      M=M+1                                                             SORB1190
      IF (ABS(PSI).LE.EPS) GO TO 54                                     SORB1191
      IF (PSI.LT.0.0) R=R+RMAX/(2.0**(M-1))                             SORB1192
      IF (PSI.GT.0.0) R=R-RMAX/(2.0**(M-1))                             SORB1193
      IF (R.LT.0.0001) GO TO 54                                         SORB1194
      IF (M.LT.20) GO TO 52                                             SORB1195
   54 XX(J)=X                                                           SORB1196
      RR(J)=R/BETA                                                      SORB1197
      IF (J.EQ.7) GO TO 200                                             SORB1198
   53 X=X+DELX                                                          SORB1199
      IF (X-XFINAL-1.0E-05) 55,55,56                                    SORB1200
   56 IF (J.EQ.0) GO TO 4                                               SORB1201
  200 WRITE (6,728) (XX(N),N=1,J)                                       SORB1202
      WRITE (6,729) (RR(N),N=1,J)                                       SORB1203
      WRITE (6,734) (RP(N),N=1,J)                                       SORB1204
      JJ=J                                                              SORB1205
      J=0                                                               SORB1206
      IF (JJ-7) 4,53,53                                                 SORB1207
      END                                                               SORB1208
      SUBROUTINE INVERS(A,NSYS,N,NMAX,MMAX)                             SORB2   1
C     LEVEL 2,A
C     SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS                        SORB2   2
C                                                                       SORB2   3
      DIMENSION A(NMAX,MMAX),X(150)                                     SORB2   4
      SIGN=1.0                                                          SORB2   5
      NPI=N+1                                                           SORB2   6
```

```
      NMI=N-1
      NPLSY=N+NSYS
      DO 14 I=1,NMI
      IPI=I+1
      MAX=I
      AMAX=ABS(A(I,I))
      DO 10 K=IPI,N
      AKMAX=ABS(A(K,I))
      IF(AKMAX.LE.AMAX) GO TO 10
      MAX=K
      AMAX=AKMAX
   10 CONTINUE
      IF(AMAX.LT.1.0E-12) GO TO 16
      IF(MAX.EQ.I) GO TO 12
      DO 11 L=I,NPLSY
      TEMP=A(I,L)
      A(I,L)=A(MAX,L)
   11 A(MAX,L)=TEMP
      SIGN=-SIGN
   12 DO 14 J=IPI,N
      IF (A(J,I)) 30,14,30
   30 CONST=-A(J,I)/A(I,I)
      DO 13 L=I,NPLSY
   13 A(J,L)=A(J,L)+A(I,L)*CONST
   14 CONTINUE
      DET=0.0
      TEMP=1.0
      DO 15 I=1,N
      IF (A(I,I)) 15,16,15
   15 CONTINUE
      GO TO 18
   16 WRITE(6,100)
  100 FORMAT(5X,18HMATRIX IS SINGULAR)
      STOP
   18 DO 21 I=NPI,NPLSY
      DO 20 KK=1,N
      K=NPI-KK
      X(K)=A(K,I)
      IF(K.EQ.N) GO TO 20
      J=K
   19 J=J+1
      X(K)=X(K)-A(K,J)*X(J)
      IF(J.NE.N) GO TO 19
   20 X(K)=X(K)/A(K,K)
      DO 21 J=1,N
   21 A(J,I)=X(J)
      RETURN
      END
      SUBROUTINE SHAPE (X,NS,XE,C,R,DRDX)
C     SUBROUTINE TO CALCULATE SHAPE
C
      DIMENSION XE(15),C(15,7)
      DO 1 K=1,NS
      XL=XE(K)
      J=K
      IF (X.LE.XL) GO TO 2
    1 CONTINUE
    2 R=C(J,1)+X*C(J,5)+X*X*C(J,6)
      ARG=X*X*C(J,2)+X*C(J,3)+C(J,4)
      DRDX=C(J,5)+2.0*X*C(J,6)
      IF (ARG.LE.0.0) RETURN
      R=R+SQRT(ARG)*C(J,7)
      DRDX=DRDX+(2.0*X*C(J,2)+C(J,3))/(2.0*SQRT(ARG))*C(J,7)
```

```
      RETURN                                                            SORB3 16
      END                                                               SORB3 17
      SUBROUTINE GEOMET (J,NTYPE,COEF,AL)
C
C     THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR THE NIELSEN
C     GENERAL POLYNOMIAL EQUATION FROM THE USUALLY KNOWN GEOMETRICAL
C     DATA FOR EACH TYPE OF BODY SEGMENT.
C
C     J IS THE NUMBER OF THE BODY SEGMENT,1<J<15,SET IN DO LOOP
C     NTYPE DESIGNATES THE TYPE OF SHAPE FOR EACH BODYSEGMENT,IN
C     SEQUENCE, FROM THE NOSE TO BASE;
C             1=OGIVE(CIRCULAR ARC) SECTION
C             2=CONICAL NOSE OR FRUSTUM SECTION
C             3=CYLINDRICAL SECTION
C
C     AL IS THE BODY LENGTH USED IN DEFINING THE COEFFICIENTS
C
      DIMENSION COEF(15,7)
      GO TO (10,20,30) NTYPE
C
   10 READ(5,*) XO,RO,CRO
C
C     THIS BRANCH OF THE SUBROUTINE HANDLES THE OGIVE SECTIONS
C     THE INPUT DATA FOR THIS CASE IS;
C        XO=X-LOCATION OF THE CENTER OF THE CIRCULAR ARC
C        RO=R-LOCATION OF THE CENTER OF THE CIRCULAR ARC
C        CRO=RADIUS OF THE CIRCULAR ARC
C
      COEF (J,1) = RO/AL
      COEF (J,2) = -1.0
      COEF (J,3) = 2.0*XO/AL
      COEF (J,4) =(CRO/AL)**2 - (XO/AL)**2
      COEF (J,5) = 0.0
      COEF (J,6) = 0.0
      IF (RO.EQ.0) THEN
         COEF (J,7)=1.0
      ELSE
         COEF (J,7)=-RO/ABS(RO)
      ENDIF
      RETURN
C
   20 READ (5,*) XI,RI,XF,RF
C
C     THIS BRANCH OF THE SUBROUTINE HANDLES THE CONICAL SECTIONS
C     THE INPUT DATA FOR THIS CASE IS;
C        XI=X-LOCATION OF THE UPSTREAM END OF THE SEGMENT
C        RI=BODY RADIUS AT THE UPSTREAM END OF THE SEGMENT
C        XF=X-LOCATION OF THE DOWNSTREAM END OF THE SEGMENT
C        RF=BODY RADIUS AT THE DOWNSTREAM END OF THE SEGMENT
C     NOTE: IF THE SEGMENT IS A CONICAL NOSE, THE VALUES OF BOTH
C           XI AND RI ARE 0.0, BUT MUST BE ENTERED
C
      COEF (J,1) = (RI*XF - RF*XI)/(AL*(XF-XI))
      DO 21 K=2,4
   21 COEF (J,K) =0.0
      COEF (J,5) = (RF-RI)/(XF-XI)
      COEF (J,6) =0.0
      COEF (J,7) =0.0
      RETURN
   30 READ (5,*) RCYL
C
C     THIS BRANCH OF THE SUBROUTINE HANDLES CYLINDRICAL SECTIONS
C
```

```
C      THE INPUT DATA FOR THIS CASE IS:
C        RCYL = THE RADIUS OF THE CYLINDER
C
       COEF (J,1) = RCYL/AL
       DO 31 K=2,7
   31  COEF (J,K) =0.0
       RETURN
       END
$BEND
```

APPENDIX A.6

Subsonic Source Program CSS Listing;
NEARSOR .CSS

```
***** LOADS AND RUNS SUBSONIC SOURCE PROGRAM *****
L #0
AS 5,#1.INP
XAL #0.OPT,IN,132
AS 6,#0.OPT
XAL #0.SRC,IN,132
AS 7,#0.SRC
START
$EXIT
```

APPENDIX B

SUBSONIC TRAJECTORY COMPUTER PROGRAM DEFINITIONS AND LISTINGS

APPENDIX B.1

SUBSONIC TRAJECTORY PROGRAM .SRC INPUT FILE DEFINITIONS

Definition of input data in the .SRC files as illustrated in Appendix A.4 is as follows:

NSECT	is the number of segments which compose the body, including simulated wakes for blunt based bodies.
X/L	are the end points of the body segments.
Lines 3 to 2+NSECT	give (a) the body segment number and (b) the seven (7) polynomial coefficients which define that body segment. There is a line of print for each of the body segments.
NSORC	is the number of sources calculated by the Subsonic Source program to represent the volume effects of the body.
X/L	is the axial location along the body where the sources are located.
Q	is the strength of the sources located at the corresponding X/L.

APPENDIX B.2

SUBSONIC TRAJECTORY PROGRAM NAMELIST INPUT DATA FILE DEFINITIONS

HEAD1 HEAD2 HEAD3	Three lines of alphanumeric text of the user's choice to define the problem, configuration, etc.
GAMF	Dispenser missile flight path angle, degrees.
FMACH	Free stream mach number.
RHO	Static air density at flight altitude or wind tunnel test conditions, slugs per cubic foot.
VINF	Free Stream velocity, feet per second.
NFU	Dispenser missile present? NFU = 0, No; NFU = 1, Yes.
NSTRS	Number of stores present, $0 \leq$ NSTRS ≤ 10.
NEJECT	Number of the store which is being separated; must = 1 for Namelist/Interactive.
FLTHC	Dispenser missile actual fuselage length, feet.
FRMAX	Dispenser missile maximum fuselage radius, feet.
SLTHC	Submissile actual length, feet.
SRMAX	Submissile maximum radius, feet.
PHI	Submissile roll angle relative to inertial system, degrees.
PSI	Submissile yaw angle relative to inertial system, degrees.
NSEG	Number of equal length segments the submissile body is divided into for the force calculation; NSEG ≤ 40.
NSEGXO	Number of submissile body segments to the flow separation location for non-linear cross flow force calculations.
NGAM	Trajectory to simulate wind tunnel captive-store trajectory? NGAM = 0, No; NGAM = 1, Yes. Inactive for single parameter sweep calculations.
NROLL	Rolling moment to be calculated? NROLL = 0, No; NROLL = 1, Yes.
NEMP	Empennage present? NEMP = 0, No; NEMP = 1, Yes.

NDAMP	Damping to be included in force calculation? NDAMP = 0, No; NDAMP = 1, Yes.
SMASS	Submissile mass, slugs (arbitrary for single variable sweep).
FIXX	I_{XX} moment of inertia, slug-ft^2 (arbitrary for single variable sweep).
FIYY	I_{YY} moment of inertia, slug-ft^2 (arbitrary for single variable sweep).
FIZZ	I_{ZZ} moment of inertia, slug-ft^2 (arbitrary for single variable sweep).
FIYZ	I_{YZ} product of inertia, slug-ft^2 (arbitrary for single variable sweep).
FIXZ	I_{XZ} product of inertia, slug-ft^2 (arbitrary for single variable sweep).
FIXY	I_{XY} product of inertia, slug-ft^2 (arbitrary for single variable sweep).
XMOM	Location along submissile axis about which the pitching and yawing moments are to be taken, negative behind nose, feet; same point about which moments of inertia are taken.
XBAR	X location of submissile C.G., measured from the moment center, feet; positive forward.
YBAR	Y location of submissile C.G., measured from the submissile axis, feet; positive to the right.
ZBAR	Z location of submissile C.G., measured from the submissile axis, feet; positive below.
CA	Submissile axial force coefficient; reference area is submissile maximum cross-sectional area.
IPLNR	Cruciform empennage, IPLNR = 0 Planar empennage, IPLNR = 1
MSF	Number of spanwise control points on each fin for force calculation; <u>must</u> be odd and $5 \leq$ MSF ≤ 11.
XTAIL	X location of tail fin leading edge-body radius juncture from submissile nose, feet; negative number.
RADAV	Average submissile body radius in empennage region, feet; positive number.

FINSS	Tail fin semispan, measured from submissile body longitudinal axis, feet; positive number.
FINROL	Initial fin orientation, degrees, 0 $\leq$ FINROL $\leq$ 90; FINROL = 0 if fins are vertical and horizontal.
CROOT	Tail fin root chord, feet; positive number.
CTIP	Tail fin tip chord, feet; positive number.
SWPLE	Tail fin leading edge sweep angle, degrees. Measured from normal to submissile longitudinal axis; sweepback is positive.
VXZERO	Submissile initial longitudinal velocity with respect to the dispenser missile, feet per second; positive forward.
VYZERO	Submissile initial lateral velocity with respect to the dispenser missile, feet per second; positive to right.
VZZERO	Submissile initial vertical velocity with respect to the dispenser missile, feet per second; positive down.
VAR(4)	Submissile initial roll rate, radians/sec.
VAR(5)	Submissile initial pitch rate, radians/sec.
VAR(6)	Submissile initial yaw rate, radians/sec.
VAR	VAR(1) to VAR(12), values from a trajectory calculation at time t to restart the trajectory calculation. The following table gives the notation used to identify VAR(1) through VAR(12) on the trajectory output.

PROGRAM NOTATION	OUTPUT NOTATION
VAR(1)	DXF, ft/sec
VAR(2)	DYF, ft/sec
VAR(3)	DZF, ft/sec
VAR(4)	P, radians/sec
VAR(5)	Q, radians/sec
VAR(6)	R, radians/sec
VAR(7)	XF of XMOM, ft
VAR(8)	YF of XMOM, ft
VAR(9)	ZF of XMOM, ft
VAR(10)	PSI, degrees
VAR(11)	THETA, degrees
VAR(12)	PHI, degrees

APPENDIX B.3

NAMELIST INPUT FILE FOR THE DISPENSER MISSILE
AND THE TWO CALIBER OGIVE NOSE SUBMISSILE

```
HEAD1='SUBSONIC TRAJECTORY.....SINGLE PARAMETER SWEEP',
HEAD2='....DISPENSER SHAPE FROM "ASUBMIS" DATABASE...INPUT: DISPENSX',
HEAD3='....SUBMISSILE SHAPE HAS 2 CAL. OGIVE NOSE...INPUT: SUBMISAX',
GAMF= 0.0,
FMACH= 0.8,
RHO= 0.001760,
VINF= 849.7,
NFO=1,
NSTRS=1,
NEJECT = 1,
FLTHC= 2.96833,
FRMAX= 0.156667,
SLTHC= 0.466083,
SRMAX= 0.038833,
NSEG=30,
NSEGXO=4,
NGAM=0,
NROLL=0,
NEMP=0,
NDAMP= 0,
SMASS= 1.0,
FIXX= 1.0,
FIYY= 1.0,
FIZZ= 1.0,
FIXY= 1.0,
FIXZ= 1.0,
FIYZ= 1.0,
XMOM= -0.2330,
XBAR=0.0,
YBAR=0.0,
ZBAR=0.0,
CA= 0.2,
VXZERO=0.0,
VYZERO=0.0,
VZZERO= 0.0,
VAR(4)=0.0,
VAR(6)=0.0,
VAR(5)=0.07 .
```

APPENDIX B.4

SUBSONIC TRAJECTORY PROGRAM SCREEN CUES AND INTERACTIVE INPUT DATA

CUE: Enter Run Mode
1 is an Alpha Sweep
2 is a Computed Trajectory
3 is a User Selected Variable Sweep
4 is an X Sweep
5 is a Z Sweep
READ (4,*) Mode

CUE: Enter Dispenser Missile Angle of Attack, Degrees
READ (4,*) ALFAC

CUE: Enter X-Location of Store Moment Center
Measured from the Dispenser Nose, Feet
READ (4,*) XSMC(1)

NOTE: XSMC(1) is positive aft of dispenser missile nose.

CUE: Enter Y-Location of Store Moment Center
Measured from the Dispenser Centerline, Feet
READ (4,*) YSMC(1)

CUE: Enter Z-Location of Store Moment Center
Measured from the Dispenser Centerline, Feet
READ (4,*) ZSMC(1)

NOTE: ZSMC(1) is positive below the dispenser missile.

CUE: Enter Store Angle of Incidence
Relative to Dispenser Centerline, Degrees
READ (4,*) SIC(1)

NOTE: SIC(1) is positive, submissile nose up towards dispenser missile centerline.

If the calculation is an angle of attack sweep, the following two cues appear:

CUE: Enter Final Value of Store Angle of Incidence,
Degrees
READ (4,*) FV

CUE: Enter Incremental Value of Store
Angle of Incidence
READ (4,*) VI

If the calculation is a physical separation trajectory, the following three cues appear:

```
CUE:  Enter Initial Time, Seconds
      READ (4,*) TIMEI

CUE:  Enter Final Time, Seconds
      READ (4,*) TIMEF

CUE:  Enter Time Increment, Seconds
      READ (4,*) DTIME
```

If the calculation is a longitudinal (X) sweep, the following two cues appear:

```
CUE:  Enter Final X-Location of Store Moment Center,
      Feet
      READ (4,*) FV

CUE:  Enter Incremental Value of Store X-Translation,
      Feet
      READ (4,*) VI
```

If the calculation is a vertical (Z) sweep, the following two cues appear:

```
CUE:  Enter Final Z-Location of Store Moment Center,
      Feet
      READ (4,*) FV

CUE:  Enter Incremental Value of Store Z-Translation,
      Feet
      READ (4,*) VI
```

If a user selected variable sweep is chosen, the following three cues appear. They define program sweep capabilities and request the appropriate data.

```
CUE:  NUMBER     VARIABLE
       8         YSMC.....Y-Location of Store Moment Center
      10         PSI.....Yaw Angle of Store
      12         PHI.....Roll Angle of Store
      Enter Number of Selected Variable
      READ (4,*) NV

CUE:  Enter Final Value of Variable
      READ (4,*) RV

CUE:  Enter Variable Increment
      READ (4,*) VI
```

NOTE: The interactive input for subsequent runs is menu driven so that only the variables that change must be reentered.

APPENDIX B.5

SUBSONIC TRAJECTORY PROGRAM
FORTRAN LISTING
NEARSUB.FTN

```
$BATCH
C      PROGRAM TRAJEC (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)            RDNCF  1
C                                                                        6DB01  2
       PROGRAM TO CALCULATE SIX-DEGREE-OF-FREEDOM STORE TRAJECTORIES     6DB01  3
               AT SUBSONIC SPEEDS                                        6DB01  4
                                                                         6DB01  5

      CHARACTER*16  NMLF
      CHARACTER*80  HEAD1,HEAD2,HEAD3

C       DIMENSION STATEMENTS                                             6DB01  6
C                                                                        6DB01  7
      DIMENSION  BYY(50,7),FVN(10,11),HEAD(20)

C       COMMON STATEMENTS                                                6DB01 29
C                                                                        6DB01 30
      COMMON /CFORCE/ DC(3,3),DX,UT(81),VAR(12),VT(81),WT(81),           6DB01 35
     1                ERAD(81),EDRDX(81),VSTORE,XMOM,ESTRMX,CNBY,CYBY,   6DB01 36
     2   CLMBY,CLNBY,DELX,CNX(40),CYX(40),CNSB,CYSB,CLMSB,CLNSB,CNCF,    6DB01 37
     3CYCF,CLMCF,CLNCF                                                   6DB01 38
      COMMON /CONSTS/ DTR,RAD,RTD                                        6DB01 41
      COMMON /CONTROL/ INP,MODE,NEJECT,NFU,NPTS,NSTRS,NV
      COMMON /COUTPT/ DVAR(12),ESTLGC,EXST(81),TIME                      6DB01 31
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                 6DB01 34
      COMMON /INDEX/ NCW,MSW,M,IMAX,IP(3),NCP(3),MSP(3),MP(3),MP1(3),    MPYLN 40
     1              MMP(3),KMAX(3),NCWS,MS,NCPS(3),MPS(3)                MPYLN 41
      COMMON /FUSDATA/ FLTHI,FRMAX,NFSOR,FXL(101),FSOR(101),NFPOLY,      6DB01 46
                       FXEND(15),FCOEF(15,7)
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,FINROL,MSF,IPLNR,CLALPH          6DB01 50
      COMMON /EFORCE/ CNEM,CLMEM,CYEM,CLNEM,CLLEM                        6DB01 51
      COMMON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,YBASEI, 6DB01 52
                     ZBASEI                                              6DB01 53
      COMMON /STRDATA/ SLTHC(10),SRMAX(10),XSMC(10),YSMC(10),ZSMC(10),
     1                 SIBCR(10),PHI(10),PSI(10),NSPOLY,SXEND(7),SIC(10),
     2                 SCOEF(7,7),NSSOR(10),SSOR(101,10),SXL(101,10)

      NAMELIST /INPUT/ HEAD1,HEAD2,HEAD3,GAMF,FMACH,RHO,VINF,NFU,
     1                 NSTRS,NEJECT,FLTHC,FRMAX,SLTHC,SRMAX,PHI,PSI,
     2                 NSEG,NSEGXO,NGAM,NROLL,NEMP,NDAMP,SMASS,FIXX,
     3                 FIYY,FIZZ,FIYZ,FIXZ,FIXY,XMOM,XBAR,YBAR,ZBAR,
     4                 CA,IPLNR,MSF,XTAIL,RADAV,FINSS,FINROL,CROOT,
     5                 CTIP,SWPLE,VXZERO,VYZERO,VZZERO,VAR
```

```
                                                                       6DB01 63
C     FORMAT STATEMENTS                                                6DB01 64
                                                                       6DB01 65
  701 FORMAT(10I5)                                                     6DB01 66
  702 FORMAT(//43X,53(1H*)/43X,53H* SUBSONIC SIX-DEGREE-OF-FREEDOM TRAJE6DB01 67
     1CTORY PROGRAM */43X,53(1H*)//)
  703 FORMAT(20A4)                                                     6DB01 68
  704 FORMAT(10X,20A4)                                                 6DB01 69
  706 FORMAT(8F10.0)                                                   6DB01 71
  707 FORMAT(///10X,17HFLIGHT CONDITIONS/15X,13HMACH NUMBER =,F5.2,    6DB01 72
     1/15X,26HFREE STREAM MASS DENSITY =,F10.7,21H SLUGS PER CUBIC FOOT,6DB01 73
     2/15X,22HFREE STREAM VELOCITY =,F8.2,16H FEET PER SECOND/15X,19HFLI6DB01 74
     3GHT PATH ANGLE =,F6.2,8H DEGREES/15X,27HDISPENSER ANGLE OF ATTACK 6DB01 75
     4=,F6.2,8H DEGREES)                                               6DB01 76
  708 FORMAT(10X,19HFUSELAGE INPUT DATA)                               6DB01 77
  709 FORMAT(15X,17HFUSELAGE LENGTH =,F10.5,5H FEET/15X,16HMAXIMUM RADIU6DB01 78
     1S =,F10.5,5H FEET/)                                              6DB01 79
  711 FORMAT(//45X,34HINCOMPRESSIBLE SOURCE DISTRIBUTION)              6DB01 81
  716 FORMAT(1H1,10X,12HSTORE NUMBER,I5,21H IS THE STORE EJECTED/)     6DB01 93


  717 FORMAT(15X,31HADDITIONAL INPUT FOR THIS STORE/20X,12HSTORE MASS =,6DB01 94
     1F10.3,6H SLUGS/20X,45HMOMENTS AND PRODUCTS OF INERTIA, SLUG - SQ FMPYLN 44
         3X,5HIXX =,F16.8/23X,5HIYY =,F16.8/23X,5HIZZ =,F16.8/23X,5HIYZ
         16.8/23X,5HIXZ =,F16.8/23X,5HIXY =,F16.8)
  718 FORMAT(20X,22HSTORE MOMENT CENTER IS,F9.5,17H FEET BEHIND NOSE/20X6DB01 98
     1,55HSTORE CENTER OF GRAVITY OFFSET FROM MOMENT CENTER, FEET/23X,6H6DB01 99
     2XBAR =,F9.5/23X,6HYBAR =,F9.5/23X,6HZBAR =,F9.5)                 6DB01100
  719 FORMAT(20X,47HPOLYNOMIALS SPECIFYING COMPRESSIBLE STORE SHAPE/   6DB01101
     123X,26HX/L OF END OF EACH SECTION/26X,7HSECTION,5X,3HX/L)        6DB01102
  720 FORMAT(28X,I2,3X,7F10.5)                                         6DB01103
  721 FORMAT(23X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTION/6DB01104
     126X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X, 6DB01105
     22HC7)                                                            6DB01106
  722 FORMAT(20X,18HSEPARATION ASSUMED,F10.5,15H FEET FROM NOSE)       6DB01107
  723 FORMAT(20X,26HAXIAL-FORCE COEFFICIENT IS,F10.5)                  6DB01108
  724 FORMAT(/20X,33HTHIS STORE HAS A PLANAR EMPENNAGE)                6DB01110
  725 FORMAT(/20X,36HTHIS STORE HAS A CRUCIFORM EMPENNAGE)             6DB01111
  726 FORMAT(20X,24HTHE EMPENNAGE FORCES ACT,F9.5,17H FEET BEHIND NOSE/ 6DB01112
     120X,50HTHE AVERAGE BODY RADIUS IN THE EMPENNAGE REGION IS,F9.5,5H 6DB01113
     2FEET)                                                            6DB01114
  727 FORMAT(20X,52HTHE TAIL FIN SEMISPAN MEASURED FROM THE BODY AXIS IS6DB01115
     1,F9.5,5H FEET/20X,29HTHE FINS ARE INITIALLY ROLLED,F6.2,41H DEGREE6DB01116
     2S FROM THE VERTICAL AND HORIZONTAL)                              6DB01117
  728 FORMAT(20X,27HTHE FIN LIFT-CURVE SLOPE IS,F9.5,11H PER RADIAN)   6DB01118
  729 FORMAT(///15X,50HPOLYNOMIALS SPECIFYING COMPRESSIBLE FUSELAGE SHAP6DB01119
     1E/18X,26HX/L OF END OF EACH SECTION/21X,7HSECTION,5X,3HX/L)      6DB01120
  730 FORMAT(23X,I2,3X,7F10.5)                                         6DB01121
  731 FORMAT(18X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTION/6DB01122
     121X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X, 6DB01123
     22HC7)                                                            6DB01124
C                                                                      6DB01126
C     CONSTANTS                                                        6DB01127
                                                                       6DB01128
      RAD=57.2957795                                                   6DB01129
      RTD = 57.29578
      ACCG=32.174                                                      6DB01130
      DTR = 0.01745329
      PI=3.1415927                                                     6DB01133

      NCASE = 0
      NDBOPT = 0

      READ (5,701,END=5) NCARDS
      WRITE (4,401)
  401 FORMAT(1X,60(1H*)/1X,'INPUT SOURCE:   FORMATTED FILE')
      INP = 2
      GO TO 1000
```

```
  5  WRITE (4,402)
402  FORMAT(1X,60(1H*)/1X,'INPUT SOURCE:    TERMINAL/NAMELIST')
     INP = 1

C       INTERACTIVE DATA INPUT

 10  CALL ACTIVINP  (IPASS,ALFAC,DTIME,TIMEI,TIMEF,FV,VI)

     IF (NCASE.GE.1) THEN
       WRITE (4,403) NMLF
403  FORMAT(/1X,'USE PREVIOUS NAMELIST FILE: ',A12,'?  YES=1,  NO=0')
       READ (4,*) IPNL
       IF (IPNL.EQ.1) GO TO 17
     ENDIF




 15  WRITE (4,405)
405  FORMAT(/1X,'ENTER NAMELIST FILE:  ( - .NAM)')
     READ (4,406) NMLF
406  FORMAT(A16)
 17  OPEN (2,FILE= NMLF,ERR=20,STATUS='OLD',FORM='FORMATTED',
    +     ACCESS='DIRECT')
     READ (2,INPUT)
     CLOSE (2)

     WRITE (6,702)
     WRITE (6,601) HEAD1,HEAD2,HEAD3
601  FORMAT(3(/10X,A80))
     GO TO 30

 20  WRITE (4,410) NMLF
410  FORMAT(/1X,'NAMELIST FILE "',A12,'" NOT AVAILABLE....TRY AGAIN')
     GO TO 15
C       FORMATTED HEADER INPUT                                          6DB01137

1000 IF (NCASE.GE.1)  READ (5,701,END=2200) NCARDS
     WRITE(6,702)                                                       6DB01143
     DO 25 J=1,NCARDS                                                   6DB01144
     READ(5,703) HEAD                                                   6DB01145
 25  WRITE (6,704) HEAD                                                 6DB01146

C       TEST CONFIGURATION

     READ (5,701)  NFU,NSTRS,NEJECT                                     6DB01170
C                                                                       6DB01148
C       FLIGHT CONDITIONS                                               6DB01149
                                                                        6DB01150
     READ (5,706) FMACH,RHO,VINF,GAMF,ALFAC                             ABIAS 22
     WRITE (6,707) FMACH,RHO,VINF,GAMF,ALFAC                            6DB01152
     BETASQ=1.0-FMACH**2                                                6DB01153
     BETA=SQRT(BETASQ)                                                  6DB01154
     ALFACR=ALFAC*DTR
     ALFAIR = ATAN(BETA*TAN(ALFACR))
     SALFFI=SIN(ALFAIR)
C                                                                       6DB01158
C      FREE STREAM COMPONENT U....FORWARDS                              6DB01159
C                            UIN..INCOMPRESSIBLE U                      6DB01160
C                            V....TO THE RIGHT LOOKING FORWARD          6DB01161
C                            W....DOWN LOOKING FORWARD                  6DB01162
C                                                                       6DB01163
     U= COS(ALFACR)                                                     6DB01164
     W= -SALFFI                                                         6DB01165
     UIN=COS(ALFAIR)

     IF (NFU.EQ.1) THEN
```

```
C       INPUT FUSELAGE DATA                                           6DB01172
C                                                                     6DB01173
      IF (INP.EQ.2)  READ (5,706) FLTHC,FRMAX                         6DB01175
      WRITE(6,708)                                                    6DB01176
      WRITE(6,709) FLTHC,FRMAX                                        6DB01178
      .THI=FLTHC/BETA                                                 6DB01179

        SRCINPUT (NCASE,0,0,NFSOR,FXL,FSOR,NFPOLY,FXEND,FCOEF,15)

       TE(6,711)                                                      6DB01184
       SOROUT (NFSOR,FXL,FSOR)                                        6DB01185
       2 N=1,NFSOR                                                    6DB01186
      XL(N)=-FLTHI*FXL(N)                                             6DB01187
      OR(N) = FSOR(N)*FLTHI**2                                        6DB01188
       ITE (6,729)                                                    6DB01194
       4 J=1,NFPOLY                                                   6DB01195
      WRITE (6,730) J,FXEND(J)                                        6DB01196
      WRITE (6,731)                                                   6DB01197
      DO 35 J=1,NFPOLY                                                6DB01198
   35 WRITE (6,730) J,(FCOEF(J,K),K=1,7)                              6DB01199
      ENDIF

      IF (NSTRS.EQ.0) GO TO 2100

C       INPUT STORE DATA                                              6DB01289
                                                                      6DB01290
      CALL STRIO(NCASE)                                               6DB01292

      CSIBCR = COS(SIBCR(NEJECT))                                     6DB01302
      SSIBCR = SIN(SIBCR(NEJECT))                                     6DB01303
      PSIR = PSI(NEJECT)*DTR
      SPSI = SIN(PSIR)
C                                                                     6DB01312
C       ADDITIONAL DATA DESCRIBING EJECTED STORE
                                                                      6DB01350
      IF (INP.EQ.1) GO TO 40
       AD (5,701) NSEG,NSEGXO,NGAM,NROLL,NEMP,NDAMP,NV
        D (5,706) SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXZ,FIXY                 6DB01352
        D (5,706) XMOM,XBAR,YBAR,ZBAR                                 6DB01353
        (NSTRS.GE.2)  WRITE (6,716) NEJECT                            6DB01354
       ITE (6,717) SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXZ,FIXY                6DB01355
      UM=ABS(XMOM)                                                    6DB01357
      WRITE (6,718) XMOM,XBAR,YBAR,ZBAR                               6DB01356
      WRITE (6,719)                                                   6DB01361
      DO 51 J=1,NSPOLY                                                6DB01362
   51 WRITE (6,720) J,SXEND(J)                                        6DB01363
      WRITE (6,721)                                                   6DB01364
      DO 52 J=1,NSPOLY                                                6DB01365
   52 WRITE (6,720) J,(SCOEF(J,K),K=1,7)                              6DB01366
C                                                                     6DB01367
C     ETERMINE GEOMETRIC PARAMETERS DESCRIBING STORE                  6DB01368
                                                                      6DB01369
       TLGC=SLTHC(NEJECT)                                             6DB01374
      TRMX=SRMAX(NEJECT)                                              6DB01375
      LOD=ESTLGC/ESTRMX
      SREF=PI*ESTRMX**2                                               6DB01376
      REFL=2.0*ESTRMX                                                 6DB01377
      DELX= ESTLGC/NSEG                                               6DB01379
      DX = .5*DELX                                                    6DB01380
      EXST(1)=0.0                                                     6DB01382
      NHSEG=2*NSEG+1                                                  6DB01383
      DO 55 J=2,NHSEG                                                 6DB01384
   55 EXST(J)=EXST(J-1)+DX                                            6DB01385
      DO 56 J=2,NHSEG,2                                               6DB01386
      XXX=EXST(J)/ESTLGC                                              6DB01387
      CALL SHAPE  (XXX,NSPOLY,SXEND,SCOEF,RR,EDRDX(J),7)              6DB01388
```

```
   56 ERAD(J)=ESTLGC*RR                                                    6DB01389
      XSEP= NSEGXO*DELX                                                    6DB01391
      WRITE(6,722) XSEP                                                    6DB01392
      IF (INP.EQ.2)  READ (5,706) CA                                       6DB01393
      WRITE (6,723) CA                                                     6DB01394
      NHSEGO=2*NSEGXO+1                                                    6DB01395
      NASYM=0                                                              6DB01396
      IF (ABS(XBAR).GT.1.0E-05) NASYM=1                                    6DB01397
      IF (ABS(YBAR).GT.1.0E-05) NASYM=1                                    6DB01398


      IF (ABS(ZBAR).GT.1.0E-05) NASYM=1                                    6DB01399
      SAPG=SIN(ALFACR+GAMF*DTR)                                            ABIAS 27
      CAPG= COS(ALFACR+GAMF*DTR)                                           ABIAS 28
C                                                                          6DB01402
C     INPUT EMPENNAGE DATA IF EMPENNAGE IS PRESENT                         6DB01403
C                                                                          6DB01404
      CNEM=0.0                                                             6DB01405
      CLMEM=0.0                                                            6DB01406
      CYEM=0.0                                                             6DB01407
      CLNEM=0.0                                                            6DB01408
      CLLEM=0.0                                                            6DB01409
      IF (NEMP.EQ.0) GO TO 57                                              6DB01410
      IF (INP.EQ.1) GO TO 50
      READ (5,701) IPLNR,NSF                                               6DB01411
      READ (5,706) XTAIL,RADAV,FINSS,FINROL,CROOT,CTIP,SWPLE               6DB01412
      EXPFSS = FINSS-RADAV
      CAVG = .5*(CROOT+CTIP)
      ASPRAT = 2.*EXPFSS/CAVG
      TAPRAT = CTIP/CROOT
      CBAR = .6666667*CROOT*(1.+TAPRAT**2/(1.+TAPRAT))
      YMAC = (1.+2.*TAPRAT)/(3.*(1.+TAPRAT))*EXPFSS
      TSWPLE = TAN(SWPLE*DTR)
      SWPMC = ATAN(TSWPLE-0.5*(CROOT-CTIP)/EXPFSS)*RAD

      CALL BRIT (FMACH,ASPRAT,TAPRAT,SWPMC,CLALFA,XCPCB)

      CLALPH = CLALFA*2.*CAVG*EXPFSS/SREF
      XTAIL = XTAIL-YMAC*TSWPLE-XCPCB*CBAR

      IF (IPLNR.EQ.1) WRITE (6,724)                                        6DB01413
      IF (IPLNR.EQ.0) WRITE (6,725)                                        6DB01414
      WRITE (6,726) XTAIL,RADAV                                            6DB01415
      WRITE (6,727) FINSS,FINROL                                           6DB01416
      WRITE (6,728) CLALPH                                                 6DB01417

      CALL EMPINI                                                          6DB01418
                                                                           6DB01419
      INITIALIZE FOR TRAJECTORY CALCULATION                                6DB01420

         VAR(10) = PSIR
         VAR(11) = SIBCR(NEJECT)
         VAR(12) = PHI(NEJECT)*DTR                                         6DB01424
      IF (INP.EQ.2)  READ (5,706) VXZERO,VYZERO,VZZERO,                    6DB01425
     1                             VAR(4),VAR(5),VAR(6)                    6DB01426
      VAR(1) = VZZERO*SSIBCR+VXZERO*CSIBCR                                 6DB01427
      VAR(2)=VYZERO                                                        6DB01428
      VAR(3) = VZZERO*CSIBCR-VXZERO*SSIBCR                                 6DB01429
      VAR(7) = -XSMC(NEJECT)                                               6DB01431
      VAR(8) = YSMC(NEJECT)                                                6DB01432
      VAR(9) = ZSMC(NEJECT)                                                6DB01433
      XNOSEI = -XSMC(NEJECT)+XNOM*CSIBCR                                   6DB01434
      YNOSEI = YSMC(NEJECT)+XNOM*SPSI                                      6DB01435
      ZNOSEI = ZSMC(NEJECT)+XNOM*SSIBCR                                    6DB01436
      XCGI=VAR(7)                                                          6DB01437
      YCGI=VAR(8)                                                          6DB01438
      ZCGI=VAR(9)                                                          6DB01439
      XBASEI = XNOSEI-SLTHC(NEJECT)*CSIBCR                                 6DB01440
      YBASEI = YNOSEI-SLTHC(NEJECT)*SPSI                                   6DB01441
      ZBASEI = ZNOSEI+SLTHC(NEJECT)*SSIBCR                                 6DB01442
```

```
      IF (NV.EQ.0) THEN
        IF (INP.EQ.2)  READ (5,706) DTIME,TIMEI,TIMEF                6DB01443
        IF (TIMEI.GT.1.E-6) THEN

        IF (INP.EQ.2)  READ (5,706) VAR
        DO 61 J=10,12
61      VAR(J) = DTR*VAR(J)
      ENDIF
      ELSE
       TIME = 0.0
       TIMEI = 0.0
       TIMEF = 0.0

      IF (INP.EQ.2)  READ (5,706) FV,VI
      IF (NV.EQ.7)  FV = -FV

      IF (NV.GE.10.AND.NV.LE.12) THEN
        FV = FV*DTR
        VI = VI*DTR
      ENDIF

      VI = ABS(VI)
      IF (VAR(NV).GT.FV)  VI = -VI
      TIMEF = 99.0
      ENDIF

      DDTIME = DTIME                                                6DB01445
      TIME=TIMEI                                                    6DB01446
      NEQ=12                                                        6DB01444
      NDIFEQ = 1                                                    6DB01451
      CALL ADAMS (DTIME,DDTIME,VAR,DVAR,NEQ,NDIFEQ,TIME,NV,FV,VI)   6DB01452

      NOUT=1                                                        6DB01453
                                                                    6DB01454
      CALCULATE AERODYNAMIC FORCES AND MOMENTS                      6DB01455
                                                                    6DB01456
      IF (NFU.EQ.1) THEN
        REFDIA = 2.*FRMAX
      ELSE
        WRITE (4,413)
413   FORMAT(1X,'ENTER REFERENCE DIAMETER')
        READ (4,*) REFDIA
      ENDIF

      NPTS= 0
62    CALL FORCE  (ALOD,CDC)

      IF(NEMP.EQ.1) CALL EMPFOR
      IF (NGAM.EQ.1) CALL DIRCOS(VAR,DC)                            6DB01459

      IF (NV.GE.7) GO TO 90

      CALCULATE ACCELERATIONS                                       6DB01461
                                                                    6DB01462
      CALCULATE COEFFICIENT MATRIX                                  6DB01463
                                                                    6DB01464
      DO 70 J=1,6                                                   6DB01465
      DO 70 K=1,6                                                   6DB01466
 70   FVN(J,K)=0.0                                                  6DB01467
      FVN(1,1)=1.0                                                  6DB01468
      FVN(2,2)=1.0                                                  6DB01469
      FVN(3,3)=1.0                                                  6DB01470
      FVN(4,4)=FIXX                                                 6DB01471
      FVN(5,5)=FIYY                                                 6DB01472
      FVN(6,6)=FIZZ                                                 6DB01473
      FVN(4,5)=-FIXY                                                6DB01474
      FVN(4,6)=-FIXZ                                                6DB01475
```

```
      FVN(5,6)=-FIYZ                                                    6DB01476
      FVN(5,4)=-FIXY                                                    6DB01477
      FVN(6,4)=-FIXZ                                                    6DB01478
      FVN(6,5)=-FIYZ                                                    6DB01479
      IF (NASYM.EQ.0) GO TO 80                                          6DB01480
      FVN(1,4)=YBAR*DC(1,3)-ZBAR*DC(1,2)                                6DB01481
      FVN(1,5)=ZBAR*DC(1,1)-XBAR*DC(1,3)                                6DB01482
      FVN(1,6)=XBAR*DC(1,2)-YBAR*DC(1,1)                                6DB01483
      FVN(2,4)=YBAR*DC(2,3)-ZBAR*DC(2,2)                                6DB01484
      FVN(2,5)=ZBAR*DC(2,1)-XBAR*DC(2,3)                                6DB01485
      FVN(2,6)=XBAR*DC(2,2)-YBAR*DC(2,1)                                6DB01486
      FVN(3,4)=YBAR*DC(3,3)-ZBAR*DC(3,2)                                6DB01487
      FVN(3,5)=ZBAR*DC(3,1)-XBAR*DC(3,3)                                6DB01488
      FVN(3,6)=XBAR*DC(3,2)-YBAR*DC(3,1)                                6DB01489
      DO 71 J=4,6                                                       6DB01490
      DO 71 K=1,3                                                       6DB01491
   71 FVN(J,K)=SMASS*FVN(K,J)                                           6DB01492
C                                                                       6DB01493
C     CALCULATE RIGHT HAND SIDE                                         6DB01494
C                                                                       6DB01495
   80 GXX=-ACCG*(SAPG*DC(1,1)-CAPG*DC(3,1))                             6DB01496
      GYY=-ACCG*(SAPG*DC(1,2)-CAPG*DC(3,2))                             6DB01497
      GZZ=-ACCG*(SAPG*DC(1,3)-CAPG*DC(3,3))                             6DB01498
      QSTORE=0.5*RHO*VSTORE**2                                          6DB01499
      QSREF=QSTORE*SREF                                                 6DB01500
      QSREFL=QSREF*REFL                                                 6DB01501
      CNORM=CNSB+CNBY+CNCF                                              6DB01502
      CSIDE=CYSB+CYBY+CYCF                                              6DB01503
      CROLL=0.0                                                         6DB01504
      CPITCH=CLMSB+CLMBY+CLMCF                                          6DB01505
      CYAW=CLNSB+CLNBY+CLNCF                                            6DB01506
      IF (NEMP.EQ.0) GO TO 81                                           6DB01507
      CNORM=CNORM+CNEM                                                  6DB01508
      CSIDE=CSIDE+CYEM                                                  6DB01509
      CROLL=CROLL+CLLEM                                                 6DB01510
      CPITCH=CPITCH+CLMEM                                               6DB01511
      CYAW=CYAW+CLNEM                                                   6DB01512
   81 CONTINUE                                                          6DB01513
      RONE=GXX-QSREF*CA/SMASS                                           6DB01515
      RTWO=GYY+QSREF*CSIDE/SMASS                                        6DB01516
      RTHR=GZZ-QSREF*CNORM/SMASS                                        6DB01517
      FVN(4,7)=QSREFL*CROLL                                             6DB01518
      FVN(5,7)=QSREFL*CPITCH                                            6DB01519
      FVN(6,7)=QSREFL*CYAW                                              6DB01520
      IF (NASYM.EQ.0) GO TO 85                                          6DB01521
      RONE=RONE+XBAR*(VAR(5)**2+VAR(6)**2)-YBAR*VAR(4)*VAR(5)-ZBAR*VAR(46DB01522
     1)*VAR(6)                                                          6DB01523
      RTWO=RTWO-XBAR*VAR(4)*VAR(5)+YBAR*(VAR(4)**2+VAR(6)**2)-ZBAR*VAR(56DB01524
     1)*VAR(6)                                                          6DB01525
      RTHR=RTHR-XBAR*VAR(4)*VAR(6)-YBAR*VAR(5)*VAR(6)+ZBAR*(VAR(4)**2+VA6DB01526
     1R(5)**2)                                                          6DB01527
      FVN(4,7)=FVN(4,7)+SMASS*(GZZ*YBAR-GYY*ZBAR)                       6DB01534
      FVN(5,7)=FVN(5,7)+SMASS*(GXX*ZBAR-GZZ*XBAR)                       6DB01535
      FVN(6,7)=FVN(6,7)+SMASS*(GYY*XBAR-GXX*YBAR)                       6DB01536
   85 FVN(1,7)=RONE*DC(1,1)+RTWO*DC(1,2)+RTHR*DC(1,3)                   6DB01537
      FVN(2,7)=RONE*DC(2,1)+RTWO*DC(2,2)+RTHR*DC(2,3)                   6DB01538
      FVN(3,7)=RONE*DC(3,1)+RTWO*DC(3,2)+RTHR*DC(3,3)                   6DB01539
      FVN(4,7)=FVN(4,7)-VAR(6)*VAR(5)*(FIZZ-FIYY)+(VAR(5)**2-VAR(6)**2)*6DB01540
     1FIYZ+VAR(4)*(VAR(5)*FIXZ-VAR(6)*FIXY)                             6DB01541
      FVN(5,7)=FVN(5,7)-VAR(6)*VAR(4)*(FIXX-FIZZ)+(VAR(6)**2-VAR(4)**2)*6DB01542
     1+VAR(5)*(VAR(6)*FIXY-VAR(4)*FIYZ)                                 6DB01543
      FVN(6,7)=FVN(6,7)-VAR(4)*VAR(5)*(FIYY-FIXX)+(VAR(4)**2-VAR(5)**2)*6DB01544
     1+VAR(6)*(VAR(4)*FIYZ-VAR(5)*FIXZ)                                 6DB01545
```

```
C                                                                       6DB01546
C     SOLVE FOR ACCELERATIONS                                           6DB01547
                                                                        6DB01548
      CALL INVERS (FVN,1,6,10,11)                                       VCORE 22
      DO 86 J=1,6                                                       6DB01550
      DVAR(J)=FVN(J,7)                                                  6DB01551
      DO 87 J=7,9                                                       6DB01552
   87 DVAR(J)=VAR(J-6)                                                  6DB01553
      SPHI=SIN(VAR(12))                                                 6DB01554
      CPHI=COS(VAR(12))                                                 6DB01555
      DVAR(10)=(VAR(5)*SPHI+VAR(6)*CPHI)/COS(VAR(11))                   6DB01556
      DVAR(11)=VAR(5)*CPHI-VAR(6)*SPHI                                  6DB01557
      DVAR(12)=VAR(4)+SIN(VAR(11))*DVAR(10)                             6DB01558
C                                                                       6DB01559
C     OUTPUT TRAJECTORY DATA IF AT END OF INTEGRATION STEP              6DB01560
C                                                                       6DB01561
   90 IF (NOUT.EQ.0) GO TO 91                                           6DB01562
      IF (NPTS.NE.0) GO TO 210

      WRITE (4,411)
  411 FORMAT(/1X,'OUTPUT LOAD AND VELOCITY DISTRIBUTIONS?',/1X,'YES=1
     1NO=0')
      READ (4,*) ILVD
      IF (NV.EQ.0) THEN
        WRITE (4,412)
  412   FORMAT(/1X,'OUTPUT COLUMNED DATA FILE?  YES=1,  NO=0')
        READ (4,*) IPLT
      ENDIF
      NPTS= NPTS+1

      CALL OUTPUT  (ILVD,IPLT,CDC,THA,CNT,CYT,CLMT,CLNT,CLLT,PHI,PSI,   6DB01563
     1THC)
      NOUT=0                                                            6DB01564

      IF (NV.NE.7.AND.NV.NE.9.AND.NV.NE.11) GO TO 180

C      CONSTRUCT BINARY OUTPUT ARRAY

      XS = -VAR(7)/REFDIA
      YS = VAR(8)/REFDIA
      ZS = VAR(9)/REFDIA
      ALPS = ALFAC+THA

      IF (NV.EQ.11) THEN
           MODE = 1
        BYY(NPTS,1) = ALPS
      ENDIF
      IF (NV.EQ.7) THEN
           MODE = 4
        BYY(NPTS,1) = XS
      ENDIF
      IF (NV.EQ.9) THEN
           MODE = 5
        BYY(NPTS,1) = ZS
      ENDIF

        BYY(NPTS,2) = CNT
        BYY(NPTS,3) = CYT
        BYY(NPTS,4) = CLMT
        BYY(NPTS,5) = CLNT
        BYY(NPTS,6) = CLLT
        BYY(NPTS,7) = CA
```

```
      (TIME+1.0E-05-TIMEF) 91,2000,2000                                6DB01565
                                                                       6DB01566
      L INTEGRATION ROUTINE                                            6DB01567
                                                                       6DB01568
 91   IF (NV .EQ. 0) GO TO 92
      NDIFEQ = 10
 92   CALL ADAMS  (DTIME,DDTIME,VAR,DVAR,NEQ,NDIFEQ,TIME,NV,FV,VI)
      IF (NDIFEQ.EQ.1) GO TO 2000                                      6DB01570
      IF (NDIFEQ.GT.7) NOUT=1                                          6DB01571
      GO TO 62                                                         6DB01572

2000  IF (NV.NE.7.AND.NV.NE.9.AND.NV.NE.11) GO TO 2100

      WRITE (4,440)
 440  FORMAT(/1X,'OUTPUT BINARY DATA BASE FILE?  YES=1, NO=0')
      READ (4,*) IFILE
      IF (IFILE.NE.1) GO TO 2100

C      OUTPUT DATA BASE FILE

      CALL DBASEOPT (NDBOPT,ALFAC,ALPS,FMACH,PHI,PSI,XS,YS,ZS,BYY)
      NDBOPT = NDBOPT+1

2100  NCASE = NCASE+1
      IF (INP.EQ.2) GO TO 1000
      WRITE (4,450)
 450  FORMAT(/1X,'DO YOU WISH TO MAKE ANOTHER RUN?  YES=1, NO=0')
      READ(4,*) IRUN
      IF (IRUN.EQ.1) GO TO 10

2200  STOP
      END                                                              6DB01573

*****************************************************************
      SUBROUTINE ACTIVINP (IPASS,ALFAC,DTIME,TIMEI,TIMEF,FV,VI)

C          SUBROUTINE INPUTS DATA FROM USER'S TERMINAL
C
      DIMENSION A(18)
      SAVE A

      COMMON /CONTROL/ INP,MODE,NEJECT,NFU,NPTS,NSTRS,NV
      COMMON /STRDATA/ SLTHC(10),SRMAX(10),XSMC(10),YSMC(10),ZSMC(10),
     1                SIBCR(10),PHI(10),PSI(10),NSPOLY,SXEND(7),SIC(10),
     2                SCOEF(7,7),NSSOR(10),SSOR(101,10),SXL(101,10)

       DTR = 0.01745329

C
      IF(IPASS.NE.0) GO TO 198
      IPASS=1
      DO 17 I=1,18
      A(I)=99.
17    CONTINUE
      WRITE(4,15)
   15 FORMAT(/1X,'ENTER RUN MODE',/5X,
     1'1 IS AN ALPHA SWEEP',/,5X,
     2'2 IS A COMPUTED TRAJECTORY',/5X,
     3'3 IS A USER SELECTED VARIABLE SWEEP',/5X,
     4'4 IS AN X SWEEP',/,5X,
     5'5 IS A Z SWEEP')
      READ (4,*) MODE
      A(1)=MODE
```

```
      WRITE(4,32)
   32 FORMAT(/1X,'ENTER DISPENSER MISSILE ANGLE OF ATTACK, DEGREES')
      READ (4,*) ALFAC
      A(2)=ALFAC

      WRITE(4,20)
   20 FORMAT(/1X,'ENTER X-LOCATION OF STORE MOMENT CENTER',
     1/1X,'MEASURED FROM THE DISPENSER NOSE, FEET')
      READ (4,*) XSMC(1)
      A(3)=XSMC(1)

      WRITE(4,25)
   25 FORMAT(1X,'ENTER Y-LOCATION OF STORE MOMENT CENTER',
     1/1X,'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      READ (4,*) YSMC(1)
      A(6)=YSMC(1)

      WRITE(4,30)
   30 FORMAT(1X,'ENTER Z-LOCATION OF STORE MOMENT CENTER',
     1/1X,'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      READ (4,*) ZSMC(1)
      A(7)=ZSMC(1)

      WRITE(4,35)
   35 FORMAT(1X,'ENTER STORE ANGLE OF INCIDENCE,',
     1/1X,'RELATIVE TO DISPENSER CENTERLINE, DEGREES')
      READ (4,*) SIC(1)
      A(10)=SIC(1)
      SIBCR(1) = SIC(1)*DTR

      GO TO (40,50,80,60,70) MODE

        =11
       ITE(4,42)
       RMAT(/1X,'ENTER FINAL VALUE OF STORE ANGLE OF INCIDENCE,',
     1' DEGREES')
      READ (4,*) FV
      A(11)=FV
      WRITE(4,44)
   44 FORMAT(1X,'ENTER INCREMENTAL VALUE OF STORE',
     1/1X,'ANGLE OF INCIDENCE')
      READ (4,*) VI
      A(12)=VI
      GO TO 115

   50 NV = 0
      WRITE (4,52)
   52 FORMAT(/1X,'ENTER INITIAL TIME, SECONDS')
      READ (4,*) TIMEI
      A(13)=TIMEI
      WRITE(4,54)
   54 FORMAT(1X,'ENTER FINAL TIME, SECONDS')
      READ (4,*) TIMEF
      A(14)=TIMEF
      WRITE(4,56)
       RMAT(1X,'ENTER TIME INCREMENT, SECONDS')
          (4,*) DTIME
        )=DTIME
        U 115

          =7
```

```
      FORMAT(/1X,'ENTER FINAL X-LOCATION OF STORE MOMENT CENTER,'
     1,' FEET')
      READ (4,*) FV
      A(4)=FV
      WRITE(4,64)
   64 FORMAT(1X,'ENTER INCREMENTAL VALUE OF STORE X-TRANSLATION,',
     1' FEET')
      READ (4,*) VI
      A(5)=VI
      GO TO 115

   70 NV=9
      WRITE(4,72)
   72 FORMAT(/1X,'ENTER FINAL Z-LOCATION OF STORE MOMENT CENTER,'
     1,' FEET')
      READ (4,*) FV
      A(8)=FV
      WRITE(4,74)
   74 FORMAT(1X,'ENTER INCREMENTAL VALUE OF STORE Z-TRANSLATION,',
     1' FEET')
      READ (4,*) VI
      A(9)=VI
      GO TO 115

      WRITE(4,82)
      FORMAT(/5X,'NUMBER',5X,'VARIABLE',/5X,6(1H-),5X,8(1H-),/8X,
     1',9X,'YSMC.....Y-LOCATION OF STORE MOMENT CENTER',/7X,
     2'10',9X,'PSI .....YAW ANGLE OF STORE',/7X,
     3'12',9X,'PHI .....ROLL ANGLE OF STORE',//1X,
     4'ENTER NUMBER OF SELECTED VARIABLE')
      READ (4,*) NV
      A(16)=NV
      WRITE(4,84)
   84 FORMAT(1X,'ENTER FINAL VALUE OF VARIABLE')
      READ (4,*) FV
      A(17)=FV
      WRITE(4,86)
   86 FORMAT(1X,'ENTER VARIABLE INCREMENT')
      READ (4,*) VI
      A(18)=VI
      GO TO 115
C
C THIS SECTION ALLOWS CHANGING OF CURRENT INTERACTIVE INPUT VALUES
C AFTER THE FIRST PASS
C
 198  WRITE(4,199)
 199  FORMAT(/1X,'ENTER 0 TO SEE MENU OF ALL INTERACTIVE INPUTS,'
     *,/1X,'OR 1 TO SEE IN SECTIONS'/)
      READ(4,*) ANS
      IF (ANS.EQ.1) GO TO 220
 200  WRITE(4,201) (A(I),I=1,18)
 201  FORMAT(/T20'1   RUN MODE..................................'F10.2
     *,        /T20'2   DISPENSER ANGLE OF ATTACK, DEG............'F10.2/
     *,        /T20'3   INITIAL X-LOCATION OF SUBMISSILE NOSE.....'F10.2
     *,        /T20'4   FINAL X-LOCATION OF SUBMISSILE NOSE.......'F10.2
     *,        /T20'5   INCREMENTAL X VALUE.......................'F10.2/
     *,        /T20'6   Y-LOCATION OF SUBMISSILE NOSE.............'F10.2/
     *,        /T20'7   INITIAL Z-LOCATION OF SUBMISSILE NOSE.....'F10.2
     *,        /T20'8   FINAL Z-LOCATION OF SUBMISSILE NOSE.......'F10.2
     *,        /T20'9   INCREMENTAL Z VALUE.......................'F10.2/
     *,        /T20'10  INITIAL SUBMISSILE ANGLE OF INCIDENCE.....'F10.2
     *,        /T20'11  FINAL VALUE OF SUBMISSILE ANGLE ..........'F10.2
     *,        /T20'12  INCREMENTAL VALUE OF SUBMISSILE ANGLE.....'F10.2/
```

```
     *,        /T20'13  INITIAL TIME, SEC........................'F10.2
     *,        /T20'14  FINAL TIME, SEC..........................'F10.2
     *,        /T20'15  TIME INCREMENT, SEC......................'F10.2/
     *,        /T20'16  NUMBER OF VARIABLE TO BE VARIED..........'F10.2
     *,        /T20'17  FINAL VALUE OF VARIATION.................'F10.2
     *,        /T20'18  INCREMENT FOR THE VARIATION..............'F10.2)
 203  WRITE(4,302)
      READ(4,*) I,AI
      GO TO 400
C
C     RUN MODE
C
 300  IF(I.NE.1) GO TO 303
      WRITE(4,301)
 301  FORMAT(/T20'1 IS AN ALPHA SWEEP',
     1        /T20'2 IS A COMPUTED TRAJECTORY',
     2        /T20'3 IS A USER SELECTED VARIABLE SWEEP',
     3        /T20'4 IS AN X SWEEP',
     4        /T20'5 IS A Z SWEEP')
      GO TO 203

      ALFAC

 303  IF(I.NE.2)GO TO 305
      WRITE(4,304)
 304  FORMAT(/T20'ENTER DISPENSER MISSILE ANGLE OF ATTACK, DEGREES')
      GO TO 203
C
C     XSMC(1)
C
 305  IF(I.NE.3)GO TO 307
      WRITE(4,306)
 306  FORMAT(/T20'ENTER INITIAL X-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER NOSE, FEET')
      GO TO 203
C
C     XFIN
C
 307  IF(I.NE.4)GO TO 309
      WRITE(4,308)
 308  FORMAT(/T20'ENTER FINAL X-LOCATION OF SUBMISSILE NOSE, FEET')
      GO TO 203
C
C     XINC
C
 309  IF(I.NE.5)GO TO 311
      WRITE(4,310)
 310  FORMAT(/T20'ENTER INCREMENTAL VALUE OF SUBMISSILE,',
     1        /T20'X-TRANSLATION, FEET')
      GO TO 203

      YSMC(1)

 311  IF(I.NE.6)GO TO 313
      WRITE(4,312)
 312  FORMAT(/T20'ENTER Y-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      GO TO 203
C
C     ZSMC(1)
C
 313  IF(I.NE.7)GO TO 315
      WRITE(4,314)
```

```
 314  FORMAT(/T20'ENTER INITIAL Z-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      GO TO 203
C
C     ZFIN
C
 315  IF(I.NE.8)GO TO 317
      WRITE(4,316)
 316  FORMAT(/T20'ENTER FINAL Z-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
C
C     ZINC
C
 317  IF(I.NE.9)GO TO 319
      WRITE(4,318)
 318  FORMAT(/T20'ENTER INCREMENTAL VALUE OF SUBMISSILE,',
     1        /T20'Z-TRANSLATION, FEET')
      GO TO 203
C
C     SIC(1)
C
 319  IF(I.NE.10)GO TO 321
      WRITE(4,320)
      FORMAT(/T20'ENTER INITIAL SUBMISSILE ANGLE OF INCIDENCE,',
              /T20'RELATIVE TO DISPENSER CENTERLINE, DEGREES')
      GO TO 203

C     ANGF
C
 321  IF(I.NE.11)GO TO 323
      WRITE(4,322)
 322  FORMAT(/T20'ENTER FINAL VALUE OF SUBMISSILE ANGLE OF INCIDENCE',
     1        /T20'RELATIVE TO DISPENSER CENTERLINE, DEGREES')
      GO TO 203
C
      ANGI

 323  IF(I.NE.12)GO TO 325
      WRITE(4,324)
 324  FORMAT(/T20'ENTER INCREMENTAL VALUE OF SUBMISSILE',
     1        /T20'ANGLE OF INCIDENCE')
      GO TO 203
C
C     TIMEI
C
 325  IF(I.NE.13)GO TO 327
      WRITE(4,326)
      FORMAT(/T20'ENTER INITIAL TIME, SECONDS')
      GO TO 203

        EF
C
 327  IF(I.NE.14)GO TO 329
      WRITE(4,328)
 328  FORMAT(/T20'ENTER FINAL TIME, SECONDS')
      GO TO 203
C
C     DTIME
C
 329  IF(I.NE.15)GO TO 331
      WRITE(4,330)
 330  FORMAT(/T20'ENTER TIME INCREMENT, SECONDS'
      GO TO 203
```

```
C
C       XNV
C
 331    IF(I.NE.16)GO TO 333
        WRITE(4,332)
 332    FORMAT(/T20'ENTER NUMBER OF VARIABLE TO BE VARIED')
        GO TO 203
C
C       VFIN
C
 333    IF(I.NE.17)GO TO 335
        WRITE(4,334)
 334    FORMAT(/T20'ENTER FINAL VALUE OF VARIATION')
        GO TO 203
C
C
 335    IF(I.NE.18)GO TO 203
        WRITE(4,336)
 336    FORMAT(/T20'ENTER INCREMENT FOR THE VARIATION')
        GO TO 203
C
C
 105    GO TO(500,510,520,530,540)MODE
 500    NV=11
        FV=A(11)
        VI=A(12)
        IF(A(11).EQ.99. .OR. A(12).EQ.99.)GOTO 550
        GO TO 110
 510    IF(A(14).EQ.99. .OR. A(15).EQ.99.)GOTO 550
        GOTO 110
 520    FV=A(17)
        VI=A(18)
        IF(A(16).EQ.99. .OR. A(17).EQ.99. .OR. A(18).EQ.99.)
       *  GOTO 550
        GOTO 110
 530    NV=7
        FV=A(4)
        VI=A(5)
        IF(A(4).EQ.99. .OR. A(5).EQ.99.)GOTO 550
        GO TO 110
 540    NV=9
        FV=A(8)
        VI=A(9)
        IF(A(7).EQ.99. .OR. A(8).EQ.99.)GO TO 550
        GO TO 110
 550    WRITE(1,410)
 410    FORMAT(//'** FINAL AND/OR INCREMENTAL VALUE
       *NOT SPECIFIED**'//)
        GO TO 203
C
 302    FORMAT(/T20'ENTER NUMBER, VALUE OR'
       1         /T20'ENTER 0,0 TO LIST CURRENT INPUTS OR'
       2         /T20'ENTER 0,1 TO RUN PROGRAM'
       3         /T20'ENTER NUMBER, 999. FOR MORE INFORMATION')
C
 400    IF(AI.EQ.999.)GO TO 300
        IF(I.EQ.0.AND.AI.EQ.0)GO TO 200
        IF(I.EQ.0.AND.AI.EQ.1)GO TO 105
        IF(I.EQ.1)MODE=AI
        IF(I.EQ.16)NV=AI
        A(I)=AI
```

```
      GO TO 203
C
C CHANGE INPUT IN STEPS INSTEAD OF WITH LARGE MENU
C
 220  WRITE(4,221)
 221  FORMAT(/5X,'ENTER 0 TO CHANGE MODE, 1 TO GO ON'/)
      READ(4,*) ANS
      IF (ANS.EQ.1) GO TO 240
      WRITE(4,222)
 222  FORMAT (/1X'MODE CODES ---')
      WRITE(4,301)
      WRITE(4,*) ('ENTER NEW MODE?')
      READ(4,*) MODE
      A(1)=MODE
C
C DISPLAY INITIAL CONDITIONS MENU
C
 240  WRITE(4,239)
 239  FORMAT (//1X'INITIAL CONDITIONS -- CURRENT STATUS'/)
      WRITE(4,241) A(2),A(3),A(6),A(7),A(10)
 241  FORMAT (/T20'1 = DISPENSER MISSILE ANGLE OF ATTACK, DEG...'F10.2
     1        /T20'2 = X-LOCATION OF SUBMISSILE NOSE............'F10.2
     2        /T20'3 = Y-LOCATION OF SUBMISSILE NOSE............'F10.2
     3        /T20'4 = Z-LOCATION OF SUBMISSILE NOSE............'F10.2
     4        /T20'5 = SUBMISSILE ANGLE OF INCIDENCE, DEG.......'F10.2/)
 244  WRITE(4, 245)
 245  FORMAT(/1X'ENTER  NUMBER,NEW VALUE  TO CHANGE AN INPUT, OR ',/1X,
     1         'ENTER  0,0  TO GO ON, OR ',/1X,
     2         'ENTER  0,1  TO SEE INITIAL VALUES AGAIN'/)
      READ(4,*) I,AI
      IF (I.EQ.0 .AND. AI.EQ.0) GO TO 250
      IF (I.EQ.0 .AND. AI.EQ.1) GO TO 240
        (I.EQ.1) THEN
        A(2)=AI
      ELSE IF (I.EQ.2) THEN
        A(3)=AI
      ELSE IF (I.EQ.3) THEN
        A(6)=AI
      ELSE IF (I.EQ.4) THEN
        A(7)=AI
      ELSE
        A(10)=AI
      ENDIF
      GO TO 244
C
C CHANGE INCREMENTS AND FINAL VALUES, IF DESIRED
C
 250  WRITE(4,251)
 251  FORMAT (/1X'ENTER 0 TO CHANGE INCREMENTS AND OTHER VALUES ',/1X
     *          'OR 1 TO RUN AGAIN'/)
      READ(4,*) ANS
      IF (ANS.EQ.1) GO TO 105
      GO TO (260,262,264,266,268) MODE
C
 260  WRITE(4, 261)
        RMAT (5X'FOR MODE = 1'/)
        TE(4,322)
        (4,*) FV
        )=FV
        TE(4,324)
      AD(4,*) VI
      A(12)=VI
      GO TO 105
```

```
      WRITE(4,263)
263   FORMAT (5X'FOR MODE = 2'/)
      WRITE(4,326)
      READ(4,*) TIMEI
      A(13)=TIMEI
      WRITE(4,328)
      READ(4,*) TIMEF
      A(14)=TIMEF
      WRITE(4,330)
      READ(4,*) DTIME
      A(15)=DTIME
      GO TO 105

264   WRITE(4,265)
265   FORMAT (5X'FOR MODE = 3'/)
      WRITE(4,332)
      READ(4,*) NV
      A(16)=NV
      WRITE(4,334)
      READ(4,*) FV
      A(17)=FV
      WRITE(4,336)
      READ(4,*) VI
      A([illegible])=VI
      GO TO 105

      WRITE(4,267)
267   FORMAT (5X'FOR MODE = 4'/)
      WRITE(4,308)
      READ(4,*) FV
      A(4)=FV
      WRITE(4,310)
      READ(4,*) VI
      A(5)=VI
      GO TO 105
C
268   WRITE(4,269)
269   FORMAT (5X'FOR MODE = 5'/)
      WRITE(4,316)
      READ(4,*) FV
      A(8)=FV
      WRITE(4,318)
      READ(4,*) VI
      A(9)=VI
      GO TO 105
C
C
110   ALFAC=A(2)
      XSMC(1)=A(3)
      YSMC(1)=A(6)
      ZSMC(1)=A(7)
      SIC(1)=A(10)
      TIMEI=A(13)
      TIMEF=A(14)
      DTIME=A(15)
      RETURN
      END

C*************************************************************************
      SUBROUTINE ADAMS (H,DS,Y,DY,NEQ,NDIFEQ,S,NV,FV,VI)
```

```
      ADAMS INTEGRATION ROUTINE

      DIMENSION Y(12),DY(12)
      DIMENSION Y1(12),DY3(12),Y5(12),E(12),Y3(12),DY1(12),Y2(12),PX(12)
     1,PF(12),DY2(12)
      DATA RTEST,ATEST,RATIO,FLB/1.42E-01,1.42E-03,100.0,1.0E-04/
      GO TO (100,200,300,400,500,600,700,800,900,950),NDIFEQ
C
C     START BY RUNGE-KUTTA
C
  100 S=S+DS
      JB=1
      DO 101 I=1,NEQ
  101 Y1(I)=Y(I)
  102 SX1=S
  103 NDIFEQ=2
      RETURN
  200 DO 201 I=1,NEQ
      DY3(I)=DY(I)
      Y5(I)=Y(I)
      TEMP=H*DY(I)
      Y(I)=0.5*TEMP+Y5(I)
  201 E(I)=TEMP
      S=0.5*H+S
      NDIFEQ=3
      RETURN
  300 DO 301 I=1,NEQ
      TEMP=H*DY(I)
      Y(I)=0.5*TEMP+Y5(I)
  301 E(I)=E(I)+2.0*TEMP
      NDIFEQ=4
      RETURN
  400 DO 401 I=1,NEQ
      TEMP=H*DY(I)
      Y(I)=Y5(I)+TEMP
  401 E(I)=E(I)+2.0*TEMP
      S=0.5*H+S
      NDIFEQ=5
      RETURN
  500 DO 501 I=1,NEQ
  501 Y(I)=(H*DY(I)+E(I))*0.16666667+Y5(I)
      GO TO (502,507,509,902),JB
  502 DO 503 I=1,NEQ
  503 Y3(I)=Y(I)
      S=S-H
 5503 H=0.5*H
      TEMP=S+H
      IF (TEMP-S) 504,504,505
  504 NDIFEQ=1
      WRITE (6,750)
  750 FORMAT(//44H  INTEGRATION INTERVAL HAS BECOME TOO SMALL./62H  WHEN
     1 IT IS ADDED TO INDEPENDENT VARIABLE, NO CHANGE RESULTS.)
      RETURN
  505 JB=2
 5505 DO 506 I=1,NEQ
  506 Y(I)=Y1(I)
      GO TO 102
  507 DO 508 I=1,NEQ
      DY1(I)=DY3(I)
  508 Y2(I)=Y(I)
      JB=3
      GO TO 103
  509 S=S-H
```

```
C
C     ERROR CHECKING
C
 5509 TEST=0.0
      DO 510 I=1,NEQ
      VX4=Y(I)
      TEMR= ABS(Y3(I)-VX4)
      IF (TEMR.LE.0.0) GO TO 510
      VX4= ABS(VX4)
      IF (VX4.LE.0.0) GO TO 512
      IF (ABS(Y3(I)).LE.0.0) GO TO 512
C
C     CHECK FOR RELATIVE ERROR
C
      IF (VX4*RTEST-TEMR) 512,511,511
  511 TEMR=TEMR/VX4
      GO TO 519
C
C     CHECK FOR ABSOLUTE ERROR
C
  512 IF (ATEST-TEMR) 514,514,513
  513 TEMR=TEMR*RATIO
      GO TO 519
C
C     BOTH TESTS FAIL, HALVE INTEGRATION INTERVAL.

        TINUE
        -H
        (JB-5) 517,514,516
         =1
      GO TO 5505
  517 DO 518 II=1,NEQ
  518 Y3(II)=Y2(II)
      GO TO 5503
  519 IF (TEST-TEMR) 520,510,510
  520 TEST=TEMR
  510 CONTINUE
C
C     OUTPUT OF RUNGE-KUTTA

      IF (JB-4) 521,802,802
  521 NDIFEQ=8
      DO 522 I=1,NEQ
      PX(I)=Y(I)
      PF(I)=DY(I)
  522 Y(I)=Y2(I)
      RETURN
  800 DO 801 I=1,NEQ
      Y(I)=PX(I)
      DY(I)=PF(I)
       (I)=DY3(I)


         Q=9
         RN
      IF (JB-5) 103,901,702
  901 CON=0.041666667*H
      JB=6
 9901 NDIFEQ=6
      GO TO 600
  902 JB=5
      TEST=FLB
      GO TO 802
```

```
C                                                                       6DB02128
C     ADAMS INTEGRATION                                                 6DB02129
C                                                                       6DB02130
  600 DO 601 I=1,NEQ                                                    6DB02131
      Y1(I)=Y(I)                                                        6DB02132
      Y(I)=Y(I)+CON*(55.0*DY(I)-59.0*DY3(I)+37.0*DY2(I)-9.0*DY1(I))     6DB02133
      DY1(I)=DY2(I)                                                     6DB02134
      DY2(I)=DY3(I)                                                     6DB02135
      DY3(I)=DY(I)                                                      6DB02136
  601 Y3(I)=Y(I)                                                        6DB02137
      S=S+H                                                             6DB02138
      NDIFEQ=7                                                          6DB02139
      RETURN                                                            6DB02140
  700 DO 701 I=1,NEQ                                                    6DB02141
  701 Y(I)=Y1(I)+CON*(9.0*DY(I)+19.0*DY3(I)-5.0*DY2(I)+DY1(I))          6DB02142
      GO TO 5509                                                        6DB02143
C                                                                       6DB02144
C     TEST FOR DOUBLING OF INTEGRATION INTERVAL                         6DB02145
                                                                        6DB02146
         (TEST-FLB) 703,9901,9901                                       6DB02147
        .,0*H                                                           6DB02148
         O 1101                                                         6DB02149
         V) = Y(NV)+VI
      .. (Y(NV) .LT. FV   .AND.  VI .GE. 0)     GO TO 951
      IF (Y(NV) .GT. FV   .AND.  VI .LT. 0)     GO TO 951
      S = 99.0
  951 RETURN
      END                                                               6DB02150

C**********************************************************************
      SUBROUTINE BESKIN (X,Y,Z,V,W)                                     6DB03  1

C     SUBROUTINE TO CALCULATE THE BESKIN UPWASH AT A FIELD POINT OF     6DB03  2
C          A CIRCULAR FUSELAGE                                          6DB03  3
                                                                        6DB03  4
      COMMON /CONSTS/ DTR,RAD,RTD                                       6DB03  5
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF
      COMMON /FUSDATA/ FLTHI,FRMAX,NFSOR,FXL(101),FSOR(101),NFPOLY,     6DB03  6
     1               FXEND(15),FCOEF(15,7)

      IF (X.GE.0.0.OR.X.LT.-FLTHI) RETURN                               6DB03  7
      XL=-X/FLTHI                                                       6DB03  8
      CALL SHAPE (XL,NFPOLY,FXEND,FCOEF,RL,DRDX,15)                     6DB03  9
      R=RL*FLTHI*BETA                                                   6DB03 10
      RS=R*R*SALFFI                                                     6DB03 11
      YS=Y*Y                                                            6DB03 12
      ZS=Z*Z                                                            6DB03 13
      YSPZS=YS+ZS                                                       6DB03 14
      TA=SQRT(YSPZS)                                                    6DB03 15
      IF (TA.LT.R) RETURN                                               6DB03 16
      YSPZSS=1.0/(YSPZS*YSPZS)                                          6DB03 17
      V=V+2.0*Y*Z*RS*YSPZSS                                             6DB03 18
      W=W-(YS-ZS)*RS*YSPZSS                                             6DB03 19
      RETURN                                                            6DB03 20
      END                                                               6DB03 21

C**********************************************************************
      SUBROUTINE BRIT (XM,AR,TR,SWEEP,CLA,XCPCB)

C     TAKEN FROM THE BRITISH DATA SHEETS  S.01.03.06
C                                         S.01.03.05
C                                         S.01.03.04
C                                         S.01.03.03
```

```
C          = FREE STREAM MACH NO.
C   AR     = ASPECT RATIO (TWO WINGS JOINED WITHOUT BODY)
C   TR     = TAPER RATIO
C   SWEEP  = WING MID-CHORD SWEEPBACK ANGLE (DEG)
C   CLA    = LIFT COEFF. SLOPE, BASED ON EXPOSED WING AREA  1/RAD.
C   XCPCB  = CENTER OF PRESSURE (PERCENT MEAN GEOMETERIC CHORD),
C            MEASURED FROM LEADING EDGE MEAN GEOMETERIC CHORD.
C
    DIMENSION XT1(15),YT1(7),ZT1(4),XT2(8),YT2(5),ZT2(4),YT4(7)
    DIMENSION BAR1(15,7,4), BAR2(8,5,4),BAR3(15,7,4),BAR4(8,7,4)
C
    DATA XT1/0.,.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,7./
    DATA YT1/0.,1.,2.,3.,4.,5.,6./
    DATA ZT1/0.,.25,.5,1./
    DATA XT2/0.,1.,2.,3.,4.,5.,6.,7./
    DATA YT2/0.,1.,2.,3.,4./
    DATA YT4/0.,1.,2.,3.,4.,5.,6./
    DATA ZT2/0.,.25,.5,1./
    DATA (((BAR1(I,J,K),I=1,15),J=1,7),K=1,2) /
   A                1.56,1.52,1.41,1.30,1.20,1.11,1.03,.95,.89,.83,
   1.78,.74,.70,.66,.63,1.56,1.52,1.40,1.27,1.17,1.08,1.00,.93,.87,
   2.82,.77,.73,.69,.65,.62,1.56,1.47,1.30,1.17,1.08,1.01,.95,.89,.84,
   3.79,.75,.71,.67,.64,.61,1.26,1.21,1.13,1.05,.99,.93,.88,.83,.79,
   4.75,.71,.68,.65,.62,.59,1.05,1.01,.98,.94,.89,.85,.81,.77,.73,.7,
   5.67,.64,.61,.59,.57,.90,.88,.86,.83,.80,.77,.74,.71,.68,.66,.63,
   6.61,.59,.57,.55,.79,.78,.77,.75,.73,.70,.68,.65,.63,.61,.59,.57,
   7.55,.53,.52,1.57,1.53,1.45,1.34,1.25,1.16,1.07,1.,.93,.875,.82,.78
   8,.735,.695,.66,1.57,1.53,1.42,1.31,1.20,1.11,1.03,.97,.91,.86,.81,
   9.765,.72,.685,.65,1.57,1.45,1.31,1.205,1.12,1.05,.98,.925,.87,.82,
   1.78,.74,.70,.665,.63,1.26,1.19,1.12,1.06,1.00,.95,.90,.85,.81,.77,
   2.74,.70,.67,.64,.61,1.05,1.00,.96,.92,.885,.86,.83,.79,.76,.725,
   3.69,.665,.64,.61,.59,.90,.855,.82,.80,.78,.76,.74,.72,.70,.67,.64,
   4.62,.60,.57,.55,.79,.75,.73,.71,.70,.68,.67,.65,.64,.62,.60,.58,
   5.56,.55,.53/
    DATA (((BAR1(I,J,K),I=1,15),J=1,7),K=3,4) /
   A                1.58,1.53,1.45,1.35,1.25,1.16,1.07,1.00,.93,.88,
   1.82,.78,.73,.69,.66,1.58,1.52,1.42,1.31,1.21,1.12,1.04,.98,.915,
   2.86,.805,.76,.72,.68,.64,1.58,1.44,1.32,1.20,1.12,1.05,.98,.92,
   3.87,.82,.78,.74,.70,.76,.63,1.26,1.20,1.13,1.06,1.01,.96,.90,.85,
   4.81,.77,.74,.70,.66,.63,.60,1.05,1.01,
   5                      .97,.93,.89,.85,.82,.78,.75,.72,.68,.66,.63,
   6.60,.58,.90,.87,.84,.81,.79,.76,.73,.71,.68,.66,.64,.62,.59,.57,
   7.55,.79,.76,.74,.71,.69,.67,.66,.64,.62,.60,.59,.57,.55,.54,.52,
   81.57,1.52,1.45,1.34,1.22,1.12,1.04,.97,.90,.84,.77,.74,.70,.67,
   9.64,1.57,1.51,1.42,1.30,1.18,1.08,1.00,.94,.88,.82,.78,.73,.69,
   1.65,.63,1.57,1.48,1.36,1.22,1.10,1.02,.95,.88,.82,.78,.73,.69,.66,
   2.63,.61,1.27,1.17,1.10,1.12,.96,.91,.86,.81,.77,.73,.69,.66,.63,
      ,.58,1.06,.99,.94,.89,.84,.81,.78,.74,.71,.68,.65,.62,.60,.58,
      ,.91,.85,.80,.77,.74,.72,.70,.68,.65,.62,.60,.58,.56,.55,.54,
      ,.74,.70,.67,.65,.63,.62,.61,.59,.58,.56,.54,.53,.52,.50/
    DATA BAR2 /
   A                .265,.270,.273,.275,.274,.273,.270,.268,.380,
   1.397,.327,.314,.303,.295,.290,.285,.500,.470,.378,.352,.335,.325,
   2.318,.313,.505,.445,.405,.380,.365,.353,.345,.340,.505,.460,.440,
   3.410,.395,.383,.375,.370,.145,.195,.225,.240,.247,.250,.250,.250,
   4.262,.265,.267,.267,.267,.267,.267,.267,.380,.332,.305,.290,.283,
   5.282,.282,.282,.395,.355,.330,.315,.405,.298,.295,.293,.405,.385,
   6.365,.345,.325,.315,.307,.303,.070,.185,.225,.237,.243,.245,.247,
   7.248,.177,.217,.235,.243,.245,.245,.245,.244,.285,.247,.243,.247,
   8.250,.250,.248,.247,.305,.267,.257,.255,.255,.255,.255,.255,.317,
   9.287,.275,.270,.270,.270,.270,.270,0.,.175,.225,.237,.243,.245,
   1.245,.245,.085,.120,.165,.205,.220,.225,.227,.230,.165,.167,.170,
```

```
1.192,.197,.205,.210,.213,.193,.190,.189,.190,.193,.197,
2              .200,.203,.200,.197,.189,.190,.193,.195,.196,.198/
 DATA (((BAR3(I,J,K),I=1,15),J=1,7),K=1,2) /
A                    1.56,1.70,1.76,1.75,1.70,1.43,1.22,1.06,.94,.85,
1.77,.71,.65,.60,.56,1.56,1.735,1.76,1.62,1.48,1.35,1.24,1.09,.96,
2.86,.78,.715,.66,.61,.565,1.56,1.535,1.47,1.395,1.31,1.22,1.14,
31.07,1.005,.90,.805,.73,.67,.62,.58,1.26,1.30,1.275,1.233,1.17,
41.11,1.05,1.00,.945,.90,.86,.775,.70,.645,.60,1.046,1.07,1.085,
51.085,1.06,1.023,.986,.95,.90,.86,.822,.79,.76,.682,.635,.985,
6.915,.93,.94,.94,.93,.91,.888,.855,.822,.788,.76,.73,.707,.682,
7.79,.795,.80,.80,.80,.802,.805,.81,.81,.78,.755,.73,.71,.68,.661,
81.57,1.56,1.85,1.82,1.64,1.41,1.21,1.06,.94,.845,.77,.71,.65,.60,
9.558,1.57,1.50,1.83,1.72,1.585,1.41,1.215,1.065,.955,.86,.78,.716,
1.66,.61,.562,1.57,1.13,1.46,1.46,1.40,1.305,1.22,1.11,.985,.89,
4,.73,.67,.62,.572,1.26,1.18,1.10,1.23,1.275,1.20,1.125,1.065,
.715,.94,.85,.772,.71,.65,.60,1.05,1.05,1.05,1.05,1.05,1.05,1.04,
.01,.962,.92,.87,.815,.74,.68,.63,.90,.908,.915,.923,.93,.942,
5.95,.935,.91,.875,.84,.815,.78,.73,.67,.79,.786,.78,.78,.779,.78,
6.785,.80,.82,.83,.82,.785,.755,.73,.704/
 DATA (((BAR3(I,J,K),I=1,15),J=1,7),K=3,4) /
A                    1.58,1.88,1.92,1.80,1.59,1.36,1.17,1.03,.92,.84,
1.75,.69,.635,.59,.55,1.58,1.80,1.90,1.76,1.55,1.355,1.18,1.045,
2.935,.847,.77,.705,.645,.60,.56,1.58,1.46,1.35,1.445,1.43,1.32,
31.20,1.08,.97,.87,.79,.72,.66,.61,.57,1.26,1.25,1.235,1.22,1.26,
41.21,1.145,1.08,.995,.91,.83,.756,.693,.64,.59,1.05,1.05,1.055,
51.055,1.055,1.07,1.05,1.015,.97,.925,.86,.80,.74,.68,.63,.90,.90,
6.902,.903,.905,.907,.91,.91,.90,.88,.86,.83,.77,.71,.67,.79,.792,
7.80,.80,.804,.81,.813,.816,.82,.822,.817,.798,.76,.74,.69,1.57,

81.79,2.0,1.8,1.52,1.29,1.12,.985,.88,.795,.72,.665,.62,.575,.535,
91.57,1.60,2.0,1.72,1.48,1.29,1.13,1.0,.90,.81,.74,.68,.63,.583,
1.545,1.57,1.55,1.52,1.48,1.44,1.29,1.16,1.03,.93,.835,.76,.70,
2.645,.60,.56,1.27,1.30,1.30,1.28,1.24,1.19,1.145,1.05,.96,.87,.80,
3.735,.68,.625,.58,1.06,1.055,1.05,1.04,1.02,.995,.98,.97,.97,.90,
4.83,.765,.70,.655,.61,.905,.90,.885,.875,.865,.86,.858,.855,.853,
5.85,.84,.785,.735,.69,.65,.795,.783,.78,.77,.762,.76,.757,.75,.75,
6 .75,.758,.76,.77,.74,.72/
 ATA BAR4 /            .250,.350,.450,.465,.475,.483,.486,.490,.380,
  2,.472,.472,.480,.487,.490,.495,.5,.5,.5,.5,.5,.501,.501,
   ,.54,.54,.54,.54,.54,.525,.520,.505,.542,.580,.580,.580,.580,
   ,.558,.560,.548,.585,.625,.625,.625,.625,.625,.565,.555,.595,
   ,.673,.673,.673,.673,.145,.295,.450,.475,.485,.490,.495,.495,
 7,.350,.437,.473,.483,.487,.492,.493,.380,.445,.445,.445,.488,
6.502,.502,.502,.395,.440,.481,.481,.481,.532,.530,.523,.405,.433,
7.470,.525,.525,.525,.570,.560,.413,.407,.503,.550,.520,.520,.520,
8.615,.425,.450,.477,.510,.560,.625,.625,.625,.070,.210,.450,.477,
9.485,.490,.493,.493,.177,.270,.365,.457,.475,.482,.485,.485,.285,
1.360,.385,.415,.480,.485,.485,.485,.305,.355,.400,.435,.465,.511,
2.510,.505,.317,.323,.326,.460,.487,.501,.593,.537,.330,.320,.310,
3.425,.505,.523,.550,.585,.337,.315,.300,.423,.500,.560,.580,.605,
40.00,.330,.445,.465,.475,.483,.485,.485,.085,.310,.400,.440,.455,
5.465,.470,.470,.165,.290,.380,.425,.448,.458,.467,.475,.193,.300,
 0,.433,.457,.470,.477,.477,.200,.330,.423,.465,.485,.493,.496,
 5,.200,.330,.423,.465,.485,.493,.496,.495,.200,.330,.423,.465,
 .485,.493,.496,.495/
 X=AR*SQRT(ABS(1.-XM**2))
 Y=AR*TAN(SWEEP/57.29578)
 Z=TR
 IF(X.GT.7.) WRITE (6,900) X
 IF(Z.GT.1.) WRITE (6,902) Z
 IF(XM.GT.1.0) GO TO 1
 CALL TABLE3 (2,CLAR,BAR1,X,XT1,15,3,Y,YT1,7,3,Z,ZT1,4,3)
 IF(Y.GT.4.) WRITE (6,901) Y
```

```
      CALL TABLE3 (2,XCPCB,BAR2,X,XT2,8,3,Y,YT2,5,3,Z,ZT2,4,3)
      CLA=CLAR*AR
      RETURN
1     CALL TABLE3 (2,CLAR,BAR3,X,XT1,15,3,Y,YT1,7,3,Z,ZT1,4,3)
      IF(Y.GT.6.) WRITE (6,901) Y
      CALL TABLE3 (2,XCPCB,BAR4,X,XT2,8,3,Y,YT4,7,3,Z,ZT2,4,3)
      CLA=CLAR*AR
      RETURN
      FORMAT (1X,'EXTRAPOLATION REQUIRED, '10HAR'BETA = F10.4)
      FORMAT (1X,'EXTRAPOLATION REQUIRED, '11HAR'TANSW = F10.4)
902   FORMAT (1X,'EXTRAPOLATION REQUIRED, TAPER RATIO = 'F10.4)
      END

C*********************************************************************
      SUBROUTINE CEL1 (RES,AK,IER)

C     ..................................................................
C
C        SUBROUTINE CEL1
C
C        PURPOSE
C           CALCULATE COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND
C
C        USAGE
C           CALL CEL1(RES,AK,IER)
C
C        DESCRIPTION OF PARAMETERS
C           RES    - RESULT VALUE
C           AK     - MODULUS (INPUT)
C           IER    - RESULTANT ERROR CODE WHERE
C                    IER=0  NO ERROR
C                    IER=1  AK NOT IN RANGE -1 TO +1
C
C        REMARKS
C           FOR AK=+1,-1 THE RESULT IS SET TO 1.E30.
C           FOR MODULUS AK AND COMPLEMENTARY MODULUS CK,
C           EQUATION AK*AK+CK*CK=1.0 IS USED.
C           AK MUST BE IN THE RANGE -1 TO +1
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           CEL1(AK)=INTEGRAL(1/SQRT((1+T*T)*(1+(CK*T)**2)), SUMMED
C           OVER T FROM 0 TO INFINITY).
C           EQUIVALENT ARE THE DEFINITIONS
C           CEL1(AK)=INTEGRAL(1/(COS(T)SQRT(1+(CK*TAN(T))**2)),SUMMED
C           OVER T FROM 0 TO PI/2),
C           CEL1(AK)=INTEGRAL(1/SQRT(1-(AK*SIN(T))**2),SUMMED OVER T
C           FROM 0 TO PI/2), WHERE K=SQRT(1.-CK*CK).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R.BULIRSCH, "NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS
C           AND ELLIPTIC FUNCTIONS", HANDBOOK SERIES SPECIAL FUNCTIONS,
C           NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C
C
      IER=0
```

```
C     TEST MODULUS
C
      GEO=1.-AK*AK
      IF(GEO)1,2,3
    1 IER=1
      RETURN
C
C     SET RESULT VALUE =OFLOW
C
    2 RES=1.E30
      RETURN
    3 GEO=SQRT(GEO)
      ARI=1.
    4 AARI=ARI
      TEST=AARI*1.E-4
      ARI=GEO+ARI
C
C     TEST OF ACCURACY
C
      IF(AARI-GEO-TEST)6,6,5
    5 GEO=SQRT(AARI*GEO)
      ARI=0.5*ARI
      GO TO 4
    6 RES=3.14159265/ARI
      RETURN
      END

C***********************************************************************
      SUBROUTINE CEL2 (RES,AK,A,B,IER)

C     ..................................................................
C
C        SUBROUTINE CEL2
C
C        PURPOSE
C           COMPUTES THE GENERALIZED COMPLETE ELLIPTIC INTEGRAL OF
C           SECOND KIND.
C
C        USAGE
C           CALL CEL2(RES,AK,A,B,IER)
C
C        DESCRIPTION OF PARAMETERS
C           RES    - RESULT VALUE
C           AK     - MODULUS (INPUT)
C           A      - CONSTANT TERM IN NUMERATOR
C           B      - FACTOR OF QUADRATIC TERM IN NUMERATOR
C           IER    - RESULTANT ERROR CODE WHERE
C                    IER=0  NO ERROR
C                    IER=1  AK NOT IN RANGE -1 TO +1
C
C        REMARKS
C           FOR AK = +1,-1 THE RESULT VALUE IS SET TO 1.E30 IF B IS
C           POSITIVE, TO -1.E30 IF B IS NEGATIVE.
C           SPECIAL CASES ARE
C           K(K) OBTAINED WITH A = 1, B = 1
C           E(K) OBTAINED WITH A = 1, B = CK*CK WHERE CK IS
C           COMPLEMENTARY MODULUS.
C           B(K) OBTAINED WITH A = 1, B = 0
C           D(K) OBTAINED WITH A = 0, B = 1
C           WHERE K, E, B, D DEFINE SPECIAL CASES OF THE GENERALIZED
C           COMPLETE ELLIPTIC INTEGRAL OF SECOND KIND IN THE USUAL
C           NOTATION, AND THE ARGUMENT K OF THESE FUNCTIONS MEANS
C           THE MODULUS.
```

```
C
C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C            NONE

          METHOD
             DEFINITION
             RES=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T))
             SUMMED OVER T FROM 0 TO INFINITY).
             EVALUATION
C            LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C            REFERENCE
C            R.BULIRSCH, "NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS
C            AND ELLIPTIC FUNCTIONS", HANDBOOK SERIES SPECIAL FUNCTIONS,
C            NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C

      IER=0

C     TEST MODULUS
C
      GEO=1.-AK*AK
      IF(GEO)1,2,6
    1 IER=1
      RETURN

      SET RESULT VALUE = OVERFLOW

         )3,5,4
         =-1.E30
         URN
         =1.E30
      TURN
    5 RES=A
      RETURN
C
C     COMPUTE INTEGRAL
C
    6 GEO=SQRT(GEO)
      ARI=1.
      AA=A
       =A+B

       =+AA*GEO
       =+W
      AA=AN
      AARI=ARI
      ARI=GEO+ARI
      AN=W/ARI+AN

      TEST OF ACCURACY

      IF(AARI-GEO-1.E-4*AARI)9,9,8
    8 GEO=SQRT(GEO*AARI)
      GEO=GEO+GEO
      GO TO 7
    9 RES=.78539816*AN/ARI
      RETURN
      END
```

....

```
      SUBROUTINE CROSS (FMACH,ALPHA,XLOD,CDCP,ETA)

C     CIRC. CYL. CROSS FLOW DRAG COEF. AND ETA (LENGTH) CORRECTION
C     ALPHA IS IN DEGREES
C
      DIMENSION TMCR(16),TCDCP(16),TLODE(11),TETA(11)
      DATA (TMCR(I), I=1,16)/ 0.0,.2,.3,.4,.5,.6,.7,.8,.9,1.0,1.1,
     1  1.3,1.5,1.7,1.9,2.0/
      DATA (TCDCP(I), I=1,16)/ 1.18,1.18,1.2,1.25,1.35,1.53,1.73,
     1  1.81,1.82,1.79,1.74,1.6,1.47,1.37,1.31,1.28/
      DATA (TLODE(I), I=1,11)/ 2.0,4.0,6.0,8.0,10.0,14.0,18.0,24.0,
     1  30.0,35.0,40.0/
      DATA (TETA(I), I=1,11)/ .565,.61,.639,.661,.681,.716,.745,
     1  .777,.795,.809,.820/

      XMCR=FMACH*SIN(ABS(ALPHA/57.29578))
      CALL INTERP (2,1,CDCP,TCDCP,XMCR,TMCR,16,3,MIN,MAX)
      XLODE=XLOD*1.18/CDCP
      CALL INTERP (2,1,ETA,TETA,XLODE,TLODE,11,3,MIN,MAX)
      RETURN
      END

C*********************************************************************
      SUBROUTINE DBASEOPT (NDBOPT,ALFAC,ALPS,FMACH,PHI,PSI,XS,YS,ZS,
     *                   BYY)

C        SUBROUTINE OUTPUTS DATA IN THE FORMAT REQUIRED
C        FOR USE WITH SUBMIS DATABASE PROGRAMS

      INTEGER*4 IC,IP
      CHARACTER*16  DBFN,ACON

      DIMENSION  BH(13),BYY(50,7)

      COMMON /CONTROL/ INP,MODE,NEJECT,NFU,NPTS,NSTRS,NV

      IF (NDBOPT.NE.0) GO TO 20

    5 WRITE (4,401)
  401 FORMAT(1X,'ENTER NAME OF DATA BASE OUTPUT FILE  ( [ ]SUBMIS.DAT
     1, IE. MSUBMIS.DAT )')
      READ (4,402) DBFN
  402 FORMAT(A16)
      OPEN(7,FILE=DBFN,ERR=10,STATUS='NEW',FORM='BINARY',RECL=56,
     1    BLOCKSIZE=56,ACCESS='DIRECT')
      GO TO 20

   10 WRITE (4,403) DBFN
  403 FORMAT(/1X,'FILE NAME "',A12,'" IS IN USE,...TRY AGAIN')
      GO TO 5

   20 WRITE (4,404)
  404 FORMAT(/1X,'ENTER TEST RUN NUMBER  (4 DIGITS , IE. 1001)')
      READ (4,*) NRUN
      WRITE (4,405)
  405 FORMAT(1X,'ENTER TEST CONFIGURATION  (MAX. 12 CHAR.)')
      READ (4,402) ACON

      RAD = SQRT(YS**2+ZS**2)

C**** QUADRANT I CALCULATION FOR PHIA ********************************

      PHIA = ATAN(YS/ZS)
      IF (YS.GE.0.AND.ZS.LE.0) GO TO 60
```

```
      IF (YS.LE.0.AND.ZS.LE.0) GO TO 30
      IF (YS.LE.0.AND.ZS.GE.0) GO TO 40
      IF (YS.GE.0.AND.ZS.GE.0) GO TO 50

C***** QUADRANT II CALCULATION FOR PHIA ******************************

   30 PHIA = 360-PHIA
      GO TO 60

C***** QUADRANT III CALCULATION FOR PHIA *****************************

   40 PHIA = 180+PHIA
      GO TO 60

C***** QUADRANT IV CALCULATION FOR PHIA ******************************

   50 PHIA = 180-PHIA

   60   BH(1) = NPTS
        BH(2) = NRUN
        BH(3) = MODE
        BH(4) = FMACH

      IF (MODE.EQ.1) THEN
                IP = Y'F00FB100'
                IC = Y'800FC000'
             BH(5) = XS
             BH(6) = YS
             BH(7) = ZS
             BH(8) = RAD
             BH(9) = PHIA
       GO TO 70
      ENDIF

      IF (MODE.EQ.4) THEN
                IP = Y'F007F100'
                IC = Y'020FC000'
             BH(5) = YS
             BH(6) = ZS
      ENDIF

      IF (MODE.EQ.5) THEN
                IP = Y'F00DF100'
                IC = Y'008FC000'
             BH(5) = XS
             BH(6) = YS
      ENDIF

             BH(7) = RAD
             BH(8) = PHIA
             BH(9) = ALPS

        BH(10) = PHI
        BH(11) = PSI
        BH(12) = ALFAC

      WRITE (7)  7,12,IC,0,IP,ACON
      WRITE (7) (BH(I), I=1,12)
      WRITE (7)
      WRITE (7)
      DO 80 I=1,NPTS
   80 WRITE (7) (BYY(I,J), J=1,7)

      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE DIRCOS (A,D)

C     SUBROUTINE TO CALCULATE DIRECTION COSINES
C
      DIMENSION A(12),D(3,3)
      SPSI=SIN(A(10))
      CPSI=COS(A(10))
      STHE=SIN(A(11))
      CTHE=COS(A(11))
      SPHI=SIN(A(12))
      CPHI=COS(A(12))
      D(1,1)=CTHE*CPSI
      D(1,2)=SPHI*STHE*CPSI-CPHI*SPSI
      D(1,3)=CPHI*STHE*CPSI+SPHI*SPSI
      D(2,1)=CTHE*SPSI
      D(2,2)=SPHI*STHE*SPSI+CPHI*CPSI
      D(2,3)=CPHI*STHE*SPSI-SPHI*CPSI
      D(3,1)=-STHE
      D(3,2)=SPHI*CTHE
      D(3,3)=CPHI*CTHE
      RETURN
      END

C*********************************************************************
      SUBROUTINE ELI1 (RES,X,CK)

C     ...............................................................
C
C        SUBROUTINE ELI1
C
C        PURPOSE
C           COMPUTES THE ELLIPTIC INTEGRAL OF FIRST KIND
C
C        USAGE
C           CALL ELI1(RES,X,CK)
C
C        DESCRIPTION OF PARAMETERS
C           RES    - RESULT VALUE
C           X      - UPPER INTEGRATION BOUND (ARGUMENT OF ELLIPTIC
C                    INTEGRAL OF FIRST KIND)
C           CK     - COMPLEMENTARY MODULUS
C
C        REMARKS
C           MODULUS K = SQRT(1.-CK*CK).
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           RES=INTEGRAL(1/SQRT((1+T*T)*(1+(CK*T)**2)), SUMMED
C           OVER T FROM 0 TO X).
C           EQUIVALENT ARE THE DEFINITIONS
C           RES=INTEGRAL(1/(COS(T)*SQRT(1+(CK*TAN(T))**2)), SUMMED
C           OVER T FROM 0 TO ATAN(X)),
C           RES=INTEGRAL(1/SQRT(1-(K*SIN(T))**2), SUMMED OVER
C           T FROM 0 TO ATAN(X)).
C           EVALUATION

C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R. BULIRSCH  NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND
C                  ELLIPTIC FUNCTIONS.
C                  HANDBOOK SERIES OF SPECIAL FUNCTIONS
C                  NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ...............................................................
```

```
C
C
      IF(X)2,1,2
    1 RES=0.
      RETURN
    2 IF(CK)4,3,4
    3 RES=ALOG(ABS(X)+SQRT(1.+X*X))
      GOTO 13
    4 ANGLE=ABS(1./X)
      GEO=ABS(CK)
      ARI=1.
      PIM=0.
    5 SQGEO=ARI*GEO
      AARI=ARI
      ARI=GEO+ARI
      ANGLE=-SQGEO/ANGLE+ANGLE
      SQGEO=SQRT(SQGEO)
      IF(ANGLE)7,6,7
C     REPLACE 0 BY SMALL VALUE
    6 ANGLE=SQGEO*1.E-8
    7 TEST=AARI*1.E-4
      IF(ABS(AARI-GEO)-TEST)10,10,8
    8 GEO=SQGEO+SQGEO
      PIM=PIM+PIM
      IF(ANGLE)9,5,5
    9 PIM=PIM+3.1415927
      GOTO 5
   10 IF(ANGLE)11,12,12
   11 PIM=PIM+3.1415927
   12 RES=(ATAN(ARI/ANGLE)+PIM)/ARI
   13 IF(X)14,15,15
   14 RES=-RES
   15 RETURN
      END

C**********************************************************************
      SUBROUTINE ELI2 (R,X,CK,A,B)

C     ..................................................................
C
C        SUBROUTINE ELI2
C
C        PURPOSE
C           COMPUTES THE GENERALIZED ELLIPTIC INTEGRAL OF SECOND KIND
C
C        USAGE
C           CALL ELI2(R,X,CK,A,B)
C
C        DESCRIPTION OF PARAMETERS
C           R      - RESULT VALUE
C           X      - UPPER INTEGRATION BOUND (ARGUMENT OF ELLIPTIC
C                    INTEGRAL OF SECOND KIND)
C           CK     - COMPLEMENTARY MODULUS
C           A      - CONSTANT TERM IN NUMERATOR
C           B      - QUADRATIC TERM IN NUMERATOR
C
C        REMARKS
C           MODULUS K = SQRT(1.-CK*CK).
C           SPECIAL CASES OF THE GENERALIZED ELLIPTIC INTEGRAL OF
C           SECOND KIND ARE
C           F(ATAN(X),K) OBTAINED WITH A=1., B=1.
C           E(ATAN(X),K) OBTAINED WITH A=1., B=CK*CK.
C           B(ATAN(X),K) OBTAINED WITH A=1., B=0.
C           D(ATAN(X),K) OBTAINED WITH A=0., B=1.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
```

```
C        METHOD
            DEFINITION
            R=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T)),
                   SUMMED OVER T FROM 0 TO X).
            EQUIVALENT IS THE DEFINITION
            R=INTEGRAL((A+(B-A)*(SIN(T))**2)/SQRT(1-(K*SIN(T))**2),
                   SUMMED OVER T FROM 0 TO ATAN(X)).
            EVALUATION
            LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R. BULIRSCH, NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND
C                  ELLIPTIC FUNCTIONS
C                  HANDBOOK SERIES OF SPECIAL FUNCTIONS
C                  NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
      ..................................................................

         TEST ARGUMENT
      IF(X)2,1,2
    1 R=0.
      RETURN
C        TEST MODULUS
    2 C=0.
      D=0.5
      IF(CK)7,3,7
    3 R=SQRT(1.+X*X)
      R=(A-B)*ABS(X)/R+B*ALOG(ABS(X)+R)
         TEST SIGN OF ARGUMENT
         +C*(A-B)
         ()5,6,6
         -R
       TURN
         INITIALIZATION
    7 AN=(B+A)*0.5
      AA=A
      R=B
      ANG=ABS(1./X)
      PIM=0.
      ISI=0
      ARI=1.
      GEO=ABS(CK)
         LANDEN TRANSFORMATION
       -AA*GEO+R
      GEO=ARI*GEO
      AA=AN
      AARI=ARI
         ARITHMETIC MEAN
      ARI=GEO+ARI
         SUM OF SINE VALUES
      AN=(R/ARI+AA)*0.5
      AANG=ABS(ANG)
      ANG=-SGEO/ANG+ANG
      PIMA=PIM
      IF(ANG)10,9,11
    9 ANG=-1.E-8*AANG
       M=PIM+3.1415927
       =ISI+1
       =ARI*ARI+ANG*ANG
       /SQRT(AANG)
       (ISI-4)13,12,12
   12 ISI=ISI-4
   13 IF(ISI-2)15,14,14
   14 P=-P
   15 C=C+P
      D=D*(AARI-GEO)*0.5/ARI
      IF(ABS(AARI-GEO)-1.E-4*AARI)17,17,16
   16 SGEO=SQRT(SGEO)
C        GEOMETRIC MEAN
```

```
      GEO=SGEO+SGEO
      PIM=PIM+PIMA
      ISI=ISI+ISI
        IO 8
C       ACCURACY WAS SUFFICIENT
   17 B=(ATAN(ARI/ANG)+PIM)*AN/ARI
      C=C+D*ANG/AANG
      GOTO 4
      END

C**********************************************************************
      SUBROUTINE EMPFOR

C     SUBROUTINE TO CALCULATE EMPENNAGE FORCES
C
      DIMENSION UTL(11,4),VTL(11,4),WTL(11,4),VN(11,4),SARG(11)

      COMMON /EFORCE/ CNEM,CLMEM,CYEM,CLNEM,CLLEM
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,FINROL,MSF,IPLNR,CLALPH
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL
      COMMON /CFORCE/ DC(3,3),DX,UT(81),VAR(12),VT(81),WT(81),
     1                ERAD(81),EDRDX(81),VSTORE,XMOM,ESTRMX,CNBY,CYBY,
     2  CLMBY,CLNBY,DELX,CNX(40),CYX(40),CNSB,CYSB,CLMSB,CLNSB,CNCF,
     3CYCF,CLMCF,CLNCF
      COMMON /CONSTS/ DTR,RAD,RTD
      COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,
     1FCONB,FCONC,CCL3(11),CCL5(11),XTAILI,DYFIN
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF

      DO 1 J=1,4
      DO 1 K=1,MSF
      UTL(K,J)=0.0
      VTL(K,J)=0.0
    1 WTL(K,J)=0.0
      XXX=XTAIL+XMOM
      JMAX=4
      IF (IPLNR.EQ.1) JMAX=2
C
C     CALCULATE VELOCITIES IN STORE COORDINATE SYSTEM
C
      DO 2 J=1,JMAX
      DO 2 K=1,MSF
      YYY=YTAIL(K,J)
      ZZZ=ZTAIL(K.J)
      CALL STTOIN (XXX,YYY,ZZZ,XI,ETA,ZETA,DC)
      XB=VAR(7)+XI
      XB=XB/BETA
      YB=VAR(8)+ETA
      ZB=VAR(9)+ZETA

      CALCULATE PERTURBATION VELOCITY FIELD

      CALL VELFLD (XB,YB,ZB,UTL(K,J),VTL(K,J),WTL(K,J))
    2 CONTINUE
C
C     CALCULATE STORE FREE-STREAM VELOCITY COMPONENTS
C
      VXI= VINF*COS(ALFACR)
      VETA=0.0
      VZETA= VINF*SIN(ALFACR)
      IF (NGAM.EQ.1) GO TO 3
      VXI=VXI+VAR(1)
      VETA=VETA+VAR(2)
      VZETA=VZETA+VAR(3)
    3 VSTORE=SQRT(VXI**2+VETA**2+VZETA**2)
      CALL INTOST (VXI,VETA,VZETA,VX,VY,VZ,DC)
C
C     ADD FREE-STREAM COMPONENTS TO PERTURBATION VELOCITIES
```

```
C     RESULTING VELOCITIES ARE POSITIVE IN -X, +Y, AND -Z DIRECTIONS
C
      VRATIO=VINF/VSTORE
      FA=VX/VSTORE
      FB=VY/VSTORE
      FC=VZ/VSTORE
      DO 4 J=1,JMAX
      DO 4 K=1,MSF
      UTL(K,J)=-VRATIO*UTL(K,J)+FA
      VTL(K,J)=+VRATIO*VTL(K,J)-FB
    4 WTL(K,J)=-VRATIO*WTL(K,J)+FC
C
C     ADD IN Q AND R DAMPING
C
      IF (NDAMP.EQ.0) GO TO 6
      DUM=-XXX/VSTORE
      FA=VAR(6)*DUM
      FB=VAR(5)*DUM
      DO 5 J=1,JMAX
      DO 5 K=1,MSF
      VTL(K,J)=VTL(K,J)+FA
    5 WTL(K,J)=WTL(K,J)+FB
C
C     DETERMINE VELOCITY COMPONENTS NORMAL TO PANELS EXCLUDING
C          ROLL DAMPING CONTRIBUTION
C
    6 DUM=-1.0
      DO 7 J=1,JMAX
      SPHI=SIN(FROLE(J))
      CPHI=COS(FROLE(J))
      DUM=-DUM
      DO 7 K=1,MSF
    7 VN(K,J)=DUM*(WTL(K,J)*SPHI-VTL(K,J)*CPHI)
C
C     CALCULATE EMPENNAGE NORMAL FORCE, SIDE FORCE, PITCHING MOMENT,
C          AND YAWING MOMENT
C
      DUM=-XTAIL/DX
      IT=DUM+0.5
      IT=IT+1
      VSO=VT(IT)
      WSO=WT(IT)
      ANG=DTR*FINROL
      WO=-WSO*COS(ANG)+VSO*SIN(ANG)
      VO=-WSO*SIN(ANG)+VSO*COS(ANG)
      A2=RADAV**2
      DO 8 K=1,MSF
      AS=(VN(K,1)+VN(K,2))/2.0+WO*A2/(RFIN(K)**2)
    8 SARG(K)=AS*CCL3(K)
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CNEM=FCONA*SUM
      CLMEM=-FCONB*SUM*FCONC
      CYEM=0.0
      CLNEM=0.0
      IF (IPLNR.EQ.1) GO TO 10
      DO 9 K=1,MSF
      BS=(VN(K,3)+VN(K,4))/2.0+VO*A2/(RFIN(K)**2)
    9 SARG(K)=BS*CCL3(K)
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CYEM=FCONA*SUM
      CLNEM=-FCONB*SUM*FCONC
   10 TA=CNEM
      TB=CYEM
      CA=COS(ANG)
      SA=SIN(ANG)
      CNM=TA*CA-TB*SA
      CYM=TA*SA+TB*CA
      TA=CLMEM
      TB=CLNEM
      CLM=TA*CA-TB*SA
```

```
      EM=TA*SA+TB*CA                                                    6DB09126
C                                                                       6DB09127
C     CALCULATE ROLLING MOMENT                                          6DB09128
C                                                                       6DB09129
      CLLEM=0.0                                                         6DB09130
      IF (NROLL.EQ.0) RETURN                                            6DB09131
      DO 11 K=1,MSF                                                     6DB09132
      ABU=(VN(K,1)-VN(K,2))/2.0                                         6DB09133
      IF (NDAMP.GT.0) ABU=ABU+VAR(4)*RFIN(K)/VSTORE                     6DB09134
      IF (IPLNR.EQ.1) GO TO 12                                          6DB09135
      ABU=ABU+(VN(K,3)-VN(K,4))/2.0                                     6DB09136
      IF (NDAMP.GT.0) ABU=ABU+VAR(4)*RFIN(K)/VSTORE                     6DB09137
   12 SARG(K)=ABU*CCL5(K)                                               6DB09138
   11 CONTINUE                                                          6DB09139
      CALL SIMSON (MSF,SARG,DYFIN,SUM)                                  6DB09140
      CLLEM=-FCONB*SUM                                                  6DB09141
      RETURN                                                            6DB09142
      END                                                               6DB09143
C*********************************************************************
      SUBROUTINE EMPINI                                                 6DB10   1
C     SUBROUTINE TO INITIALIZE FOR EMPENNAGE CALCULATION                6DB10   2
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                6DB10   6
      COMMON /CFORCE/ DC(3,3),DX,UT(81),VAR(12),VT(81),WT(81),          6DB10   7
     1                ERAD(81),EDRDX(81),VSTORE,XMOM,ESTRMX,CNBY,CYBY,  6DB10   8
     2   CLMBY,CLNBY,DELX,CNX(40),CYX(40),CNSB,CYSB,CLMSB,CLNSB,CNCF,   6DB10   9
     3CYCF,CLMCF,CLNCF                                                  6DB10  10
      COMMON /CONSTS/ DTR,RAD,RTD                                       6DB10  11
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,FINROL,MSF,IPLNR,CLALPH         6DB10  12
      COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,  6DB10  13
     1FCONB,FCONC,CCL3(11),CCL5(11),XTAILI,DYFIN                        6DB10  14
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF
C                                                                       6DB10  15
C     LOCATE SPANWISE CONTROL POINTS                                    6DB10  16
C                                                                       6DB10  17
      MS=MSF-1                                                          6DB10  18
      TB=FINSS-RADAV                                                    6DB10  19
      DYFIN=TB/TA                                                       6DB10  20
      DO 1 J=1,MSF                                                      6DB10  21
      TC=J-1                                                            6DB10  22
    1 RFIN(J)=RADAV+TC*DYFIN                                            6DB10  23
C                                                                       6DB10  24
C     LOCATE CONTROL POINTS ON FOUR FINS IN STORE COORDINATE SYSTEM     6DB10  25
C                                                                       6DB10  26
      IF (XTAIL.GT.0.0) XTAIL=-XTAIL                                    6DB10  27
      XTAILI=XTAIL/BETA                                                 6DB10  28
      FROLE(1)=DTR*(FINROL+90.0)                                        6DB10  29
      FROLE(2)=DTR*(FINROL+270.0)                                       6DB10  30
      FROLE(3)=DTR*(FINROL+180.0)                                       6DB10  31
      FROLE(4)=DTR*(FINROL)                                             6DB10  32
      JMAX=4                                                            6DB10  33
      IF (IPLNR.EQ.1) JMAX=2                                            6DB10  34
      DO 2 J=1,JMAX                                                     6DB10  35
      SPHI=SIN(FROLE(J))                                                6DB10  36
      CPHI=COS(FROLE(J))                                                6DB10  37
      DO 2 K=1,MSF                                                      6DB10  38
      YTAIL(K,J)=RFIN(K)*SPHI                                           6DB10  39
    2 ZTAIL(K,J)=-RFIN(K)*CPHI                                          6DB10  40
C                                                                       6DB10  41
C     COMPUTE CONSTANTS                                                 6DB10  42
C                                                                       6DB10  43
      FCONA=CLALPH/(3.1415927*TB*TB)                                    6DB10  44
      FCONB=FCONA/(2.0*ESTRMX)                                          6DB10  45
      FCONC=-XTAIL-XMOM                                                 6DB10  46
      SHS=FINSS**2                                                      6DB10  47
      A4=RADAV**4                                                       6DB10  48
      A2=RADAV**2                                                       6DB10  49
      TOVPI=2.0/3.1415927                                               6DB10  50
```

```
      DO 3 J=1,MSF
      R2=RFIN(J)**2
      ARG=(SHS*R2-A4)*(SHS-R2)/(SHS*R2)
    3 CCL3(J)=4.0*SQRT(ARG)
      IF (NROLL.EQ.0) RETURN
      IF (IPLNR.EQ.0) GO TO 10
C
C     COMPUTE LOADING COEFFICIENTS FOR ROLLING MOMENT CALCULATION
C
C     PLANAR EMPENNAGE
C
      TA=1.0+TOVPI*ACOS(2.0*FINSS*RADAV/(SHS+A2))
      TB=(SHS-A2)/(SHS+A2)
      DO 4 J=1,MSF
      R=RFIN(J)
      R2=R*R
      CCL5(J)=TA*(R+A2/R)*CCL3(J)/4.0
      IF (J.EQ.1.OR.J.EQ.MSF) GO TO 4
      TC=(R2+A2)*TB/(R2-A2)
      CCL5(J)=CCL5(J)+TOVPI*((R-A2/R)**2)*ALOG(TC+SQRT(TC*TC-1.0))
    4 CONTINUE
      RETURN
C
C     CRUCIFORM EMPENNAGE
C
   10 CAPRSQ=(SHS+A4/SHS)/4.0
      CTWGAM=A2/(2.0*CAPRSQ)
      TWGAM=ACOS(CTWGAM)
      STWGAM=SIN(TWGAM)
      TTWGAM=STWGAM/CTWGAM
      AK1=STWGAM
      CALL CEL1 (AKK1,STWGAM,IER)
      R2=RFIN(1)**2
      R4=R2*R2
      SH4=SHS*SHS
      TWOTHE=ACOS(SHS*(A4+R4)/(R2*(A4+SH4)))
      CCL5(1)=4.0*TOVPI*CAPRSQ*(-CTWGAM*ALOG(CTWGAM)+0.5*AKK1*SIN(2.0*
     1TWOTHE))
      CK=SQRT(1.0-AK1*AK1)
      CKCK=CK*CK
      CALL CEL2 (EPIO2K,AK1,1.0,CKCK,IER)
      DO 11 J=2,MSF
      R2=RFIN(J)**2
      R4=R2*R2
      CTWTHE=SHS*(A4+R4)/(R2*(A4+SH4))
      TWTHE=ACOS(CTWTHE)
      STWTHE=SIN(TWTHE)
      TTWTHE=TAN(TWTHE)
      ARG1=STWTHE/STWGAM
      ARG2=TTWTHE/TTWGAM
      ARG3=ASIN(ARG1)
      CCL5(J)=4.0*TOVPI*CAPRSQ*(CTWTHE*0.5*ALOG((1.0+ARG1)/(1.0-ARG1))
     1-CTWGAM*0.5*ALOG((1.0+ARG2)/(1.0-ARG2)))
      TANA1=TAN(ARG3)
      CALL ELI1 (FAK,TANA1,CK)
      CALL ELI2 (EAK,TANA1,CK,1.0,CKCK)
      CCL5(J)=CCL5(J)+2.0*TOVPI*CAPRSQ*(AKK1*SIN(2.0*TWTHE)-2.0*STWGAM
     1*COS(ARG3)*(AKK1*EAK-EPIO2K*FAK))
   11 CONTINUE
      RETURN
      END

C*********************************************************************
      SUBROUTINE FORCE (ALOD,CDC)

C     SUBROUTINE TO CALCULATE AERODYNAMIC FORCES AND MOMENTS ON BODY

      DIMENSION DUMMY(12)
```

```
      COMMON /INDEX/ NCW,MSW,M,IMAX,IP(3),NCP(3),MSP(3),MP(3),MP1(3),      MPYLN 74
                 MMP(3),KMAX(3),NCWS,MS,NCPS(3),MPS(3)                    MPYLN 75
      COMMON /CONSTS/ DTR,RAD,RTD                                         6DB11 12
      COMMON /FUSDATA/ FLTHI,FRMAX,NFSOR,FXL(101),FSOR(101),NFPOLY,        6DB11 15
                   FXEND(15),FCOEF(15,7)
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                   6DB11 17
      COMMON /CFORCE/ DC(3,3),DX,UT(81),VAR(12),VT(81),WT(81),            6DB11 18
     1               ERAD(81),EDRDX(81),VSTORE,XMOM,ESTRMX,CNBY,CYBY,     6DB11 19
     2   CLMBY,CLNBY,DELX,CNX(40),CYX(40),CNSB,CYSB,CLMSB,CLNSB,CNCF,     6DB11 20
     3CYCF,CLMCF,CLNCF                                                    6DB11 21
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF

      IF (NGAM) 20,20,21                                                  6DB11 23
   20 CALL DIRCOS(VAR,DC)                                                 6DB11 24
      GO TO 22                                                            6DB11 25
   21 DUMMY(10)= VAR(10)-ATAN(VAR(2)/(VINF*COS(ALFACR)+VAR(1)))
      DUMMY(11)= VAR(11)+ATAN(VAR(3)/(VINF*COS(ALFACR)+VAR(1)))
      DUMMY(12)=VAR(12)                                                   6DB11 28
      CALL DIRCOS(DUMMY,DC)                                               6DB11 29
   22 CONTINUE                                                            6DB11 30

      DO 1 J=1,NHSEG                                                      6DB11 31
      UT(J)=0.0                                                           6DB11 32
      VT(J)=0.0                                                           6DB11 33
    1 WT(J)=0.0                                                           6DB11 34
      XX=DX                                                               6DB11 40
      DO 10 N=1,NHSEG                                                     6DB11 41
      XX=XX-DX                                                            6DB11 43
C                                                                         6DB11 44
C     LOCATE POINT IN INCOMPRESSIBLE FUSELAGE SYSTEM                      6DB11 45
C                                                                         6DB11 46
      XXX=XX+XMOM                                                         6DB11 47
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)                            6DB11 48
      XB=VAR(7)+XI                                                        6DB11 49
      XB=XB/BETA                                                          6DB11 50
      YB=VAR(8)+ETA                                                       6DB11 51
      ZB=VAR(9)+ZETA                                                      6DB11 52
C                                                                         6DB11 53
C     CALCULATE PERTURBATION VELOCITY FIELD                               6DB11 54
C                                                                         6DB11 55
      CALL VELFLD (XB,YB,ZB,UT(N),VT(N),WT(N))                            6DB11 56
   10 CONTINUE                                                            6DB11 57
C                                                                         6DB11 58
C     CALCULATE STORE FREE-STREAM VELOCITY AND COMPONENTS                 6DB11 59
C                                                                         6DB11 60
      IF (NGAM) 56,56,55                                                  6DB11 61
   55 VXI= VINF*COS(ALFACR)
      VETA=0.0                                                            6DB11 63
      VZETA= VINF*SIN(ALFACR)
      GO TO 57                                                            6DB11 65
   56 VXI= VINF*COS(ALFACR)+VAR(1)
      VETA=VAR(2)                                                         6DB11 67
      VZETA= VINF*SIN(ALFACR)+VAR(3)
   57 VSTORE=SQRT(VXI**2+VETA**2+VZETA**2)                                6DB11 69
      CALL INTOST (VXI,VETA,VZETA,VX,VY,VZ,DC)                            6DB11 70
C                                                                         6DB11 71
C     ADD FREE-STREAM COMPONENTS TO PERTURBATION VELOCITIES               6DB11 72
C     RESULTING VELOCITIES ARE IN -X, +Y, AND -Z DIRECTIONS               6DB11 73
C                                                                         6DB11 74
      VRATIO=VINF/VSTORE                                                  6DB11 75
      TA=VX/VSTORE                                                        6DB11 76
      TB=VY/VSTORE                                                        6DB11 77
      TC=VZ/VSTORE                                                        6DB11 78
      DO 60 N=1,NHSEG                                                     6DB11 79
      UT(N)=-VRATIO*UT(N)+TA                                              6DB11 80
      VT(N)=VRATIO*VT(N)-TB                                               6DB11 81
   60 WT(N)=-VRATIO*WT(N)+TC                                              6DB11 82
C                                                                         6DB11 83
C     ADD IN DAMPING TERMS                                                6DB11 84
C                                                                         6DB11 85
```

```
      IF (NDAMP.EQ.0) GO TO 70
      XX=-DX
      DO 65 N=1,NHSEG
      XX=XX+DX
      DUM=(XX-XMOM)/VSTORE
      VT(N)=VT(N)+VAR(6)*DUM
   65 WT(N)=WT(N)+VAR(5)*DUM
C
C     CALCULATE FORCES AND MOMENTS
C
   70 CONA=2.0/(ESTRMX**2)
      ALPHA=VAR(11)*57.29578
      CALL CROSS (FMACH,ALPHA,ALOD,CDCP,ETA)
      CDC=CDCP*ETA

      CONB=CONA*CDC/3.1415927
      XSTOR=-DX
      CNBY=0.0
      CYBY=0.0
      CLMBY=0.0
      CLNBY=0.0
C
C     BUOYANCY FORCES AND MOMENTS
C
      TA=DELX/(2.0*ESTRMX)
      DO 71 N=2,NHSEG,2
      NN=N/2
      XSTOR=XSTOR+DELX
      DUM=XMOM-XSTOR
      CONC=CONA*ERAD(N)**2
      COND=DUM*TA
      DCN=CONC*(WT(N+1)-WT(N-1))/DELX
      DCY=CONC*(VT(N+1)-VT(N-1))/DELX
      CNBY=CNBY+DELX*DCN
      CYBY=CYBY+DELX*DCY
      CNX(NN)=DCN
      CYX(NN)=DCY
      CLMBY=CLMBY+COND*DCN
      CLNBY=CLNBY+COND*DCY
   71 CONTINUE
C
C     SLENDER BODY THEORY FORCES AND MOMENTS
C
      CNSB=0.0
      CYSB=0.0
      CLMSB=0.0
      CLNSB=0.0
      XSTOR=-DX
      DO 72 N=2,NHSEGO,2
      XSTOR=XSTOR+DELX
      NN=N/2
      CONC=2.0*CONA*ERAD(N)*EDRDX(N)
      COND=(XMOM-XSTOR)*TA
      DCN=CNX(NN)+CONC*WT(N)
      DCY=CYX(NN)+CONC*VT(N)
      CNSB=CNSB+DELX*DCN
      CYSB=CYSB+DELX*DCY
      CNX(NN)=CNX(NN)+DCN
      CYX(NN)=CYX(NN)+DCY
      CLMSB=CLMSB+COND*DCN
      CLNSB=CLNSB+COND*DCY
   72 CONTINUE

      VISCOUS CROSSFLOW FORCES AND MOMENTS

      CNCF=0.0
      CYCF=0.0
      CLMCF=0.0
      CLNCF=0.0
```

```
      IF (NHSEG.EQ.NHSEGO) GO TO 80
      NI=NHSEGO+1
      DO 73 N=NI,NHSEG,2
      VC=SQRT(WT(N)**2+VT(N)**2)
      XSTOR=XSTOR+DELX
      NN=N/2
      CONC=CONB*ERAD(N)*VC
      COND= 0.5*(XMOM-XSTOR)/ESTRMX*DELX
      DCN=CONC*WT(N)
      DCY=CONC*VT(N)
      CNCF=CNCF+DELX*DCN
      CYCF=CYCF+DELX*DCY
      CNX(NN)=CNX(NN)+DCN
      CYX(NN)=CYX(NN)+DCY
      CLMCF=CLMCF+COND*DCN
      CLNCF=CLNCF+COND*DCY
   73 CONTINUE
   80 CONTINUE
      RETURN
      END
      FUNCTION FX(M,THETA,RM,R)
      PI2=1.57079633
      S=M
      FX=SIN(S*THETA)*SIN(THETA-RM+PI2)-COS(S*THETA)*COS(THETA-RM+PI2)
      FX=FX*(S/(R**(M+1)))
      RETURN
      END

**********************************************************************
      SUBROUTINE INTERP (IXTRP,LMT,Y,YT,X,XT,NX,NPX,MINX,MAXX)

      DIMENSION XT(NX),YT(NX)

      IF (IXTRP.EQ.1) GO TO 110
      Y=YT(1)
      IF (X.LE.XT(1)) RETURN
      Y=YT(NX)
      IF (X.GE.XT(NX)) RETURN
  110 IF (LMT.EQ.2) GO TO 120
      CALL LIMIT (X,XT,NX,NPX,MINX,MAXX)
  120 Y=YT(MINX)
      IF (MINX.EQ.MAXX) RETURN
      Y=0.
      DO 140 J=MINX,MAXX
      P=1.
      DO 130 I=MINX,MAXX
      IF (I.EQ.J) GO TO 130
      P=P*(X-XT(I))/(XT(J)-XT(I))
  130 CONTINUE
  140 Y=Y+YT(J)*P
      RETURN

**********************************************************************
      SUBROUTINE INTOST (XI,ETA,ZETA,X,Y,Z,DC)
C
C     SUBROUTINE TO TRANSFORM FROM INERTIAL TO STORE SYSTEM
C
      DIMENSION DC(3,3)
      X=XI*DC(1,1)+ETA*DC(2,1)+ZETA*DC(3,1)
      Y=XI*DC(1,2)+ETA*DC(2,2)+ZETA*DC(3,2)
      Z=XI*DC(1,3)+ETA*DC(2,3)+ZETA*DC(3,3)
      RETURN
      END

**********************************************************************
      SUBROUTINE INVERS (A,NSYS,N,NMAX,MMAX)
C
C     SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS
C
      DIMENSION A(NMAX,MMAX),X(200)
      SIGN=1.0
```

```
      NPI=N+1                                                           6DB15   6
      NMI=N-1                                                           6DB15   7
      NPLSY=N+NSYS                                                      6DB15   8
      DO 14 I=1,NMI                                                     6DB15   9
      IPI=I+1                                                           6DB15  10
      MAX=I                                                             6DB15  11
      AMAX=ABS(A(I,I))                                                  6DB15  12
      DO 10 K=IPI,N                                                     6DB15  13
      AKMAX=ABS(A(K,I))                                                 6DB15  14
      IF(AKMAX.LE.AMAX) GO TO 10                                        6DB15  15
      MAX=K                                                             6DB15  16
      AMAX=AKMAX                                                        6DB15  17
   10 CONTINUE                                                          6DB15  18
      IF(AMAX.LT.1.0E-12) GO TO 16                                      6DB15  19
      IF(MAX.EQ.I) GO TO 12                                             6DB15  20
      DO 11 L=I,NPLSY                                                   6DB15  21
      TEMP=A(I,L)                                                       6DB15  22
      A(I,L)=A(MAX,L)                                                   6DB15  23
   11 A(MAX,L)=TEMP                                                     6DB15  24
      SIGN=-SIGN                                                        6DB15  25
   12 DO 14 J=IPI,N                                                     6DB15  26
      IF (A(J,I)) 30,14,30                                              6DB15  27
   30 CONST=-A(J,I)/A(I,I)                                              6DB15  28
      DO 13 L=I,NPLSY                                                   6DB15  29
   13 A(J,L)=A(J,L)+A(I,L)*CONST                                        6DB15  30
   14 CONTINUE                                                          6DB15  31
      DO 15 I=1,N                                                       6DB15  32
      IF (A(I,I)) 15,16,15                                              6DB15  33
   15 CONTINUE                                                          6DB15  34
      GO TO 18                                                          6DB15  35
   16 WRITE(6,100)                                                      6DB15  36
  100 FORMAT(5X,18HMATRIX IS SINGULAR)                                  6DB15  37
      STOP                                                              6DB15  38
   18 DO 21 I=NPI,NPLSY                                                 6DB15  39
      DO 20 KK=1,N                                                      6DB15  40
      K=NPI-KK                                                          6DB15  41
      X(K)=A(K,I)                                                       6DB15  42
      IF(K.EQ.N) GO TO 20                                               6DB15  43
      J=K                                                               6DB15  44
   19 J=J+1                                                             6DB15  45
      X(K)=X(K)-A(K,J)*X(J)                                             6DB15  46
      IF(J.NE.N) GO TO 19                                               6DB15  47
   20 X(K)=X(K)/A(K,K)                                                  6DB15  48
      DO 21 J=1,N                                                       6DB15  49
   21 A(J,I)=X(J)                                                       6DB15  50
      RETURN                                                            6DB15  51
      END                                                               6DB15  52

C*********************************************************************
      SUBROUTINE LIMIT (X,XT,NX,NP,MINX,MAXX)

C     THIS SUBROUTINE FINDS THE RANGE OF SUBSCRIPTS TO BE
C     CONSIDERED FOR INTERPOLATION.
C
      DIMENSION XT(NX)
      NPX=NP
      IF (NPX.GT.NX) NPX=NX
      DO 25 I=1,NX
      IF (XT(I)-X) 25,55,45
   25 CONTINUE
C     .................................. GREATER THAN MAX. SUBSCRIPT
   35 MAXX=NX
      MINX=NX-NPX+1

      RETURN
C     .................................. WITHIN RANGE
   45 MINX=I-NPX/2
      MAXX=MINX+NPX-1
      IF (MAXX.GT.NX) GO TO 35
      IF (MINX.GE.1) RETURN
```

```
C     ................................ LESS THAN MIN. SUBSCRIPT
      MINX=1
      MAXX=NPX
      RETURN
C     ................................ NO INTERPOLATION NECESSARY
   55 MINX=I
      MAXX=I
      RETURN
      END

C*****************************************************************
      SUBROUTINE OUTPUT (ILVD,IPLT,CDC,THA,CNT,CYT,CLMT,CLNT,CLLT,PHI,   6DB20  1
     *                   PSI,SLTHC)

C     SUBROUTINE TO OUTPUT FORCE AND MOMENT DATA AND TRAJECTORY DATA     6DB20  2

      DIMENSION SLTHC(10)
      CHARACTER*16  FPLOT
      CHARACTER*5  VNAME

      COMMON /CONTROL/ INP,MODE,NEJECT,NFU,NPTS,NSTRS,NV
      COMMON /COUTPT/ DVAR(12),ESTLGC,EXST(81),TIME                      6DB20  7
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                 6DB20  8
      COMMON /CFORCE/ DC(3,3),DX,UT(81),VAR(12),VT(81),WT(81),           6DB20  9
     1                ERAD(81),EDRDX(81),VSTORE,XMOM,ESTRMX,CNBY,CYBY,   6DB20 10
     2   CLMBY,CLNBY,DELX,CNX(40),CYX(40),CNSB,CYSB,CLMSB,CLNSB,CNCF,    6DB20 11
     3CYCF,CLMCF,CLNCF                                                   6DB20 12
      COMMON /EFORCE/ CNEM,CLMEM,CYEM,CLNEM,CLLEM                        6DB20 13
      COMMON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,YBASEI,6DB20 14
     1                ZBASEI                                             6DB20 15
  701 FORMAT(///5X,100(1H.)//6X,6HTIME =,F8.5,8H SECONDS)                6DB20 16
  702 FORMAT(/5X,29HFORCE AND MOMENT COEFFICIENTS/25X,2HCN,8X,2HCY,7X,   6DB20 17
     13HCLM,7X,3HCLN,7X,3HCLL/7X,8HBUOYANCY,5X,4F10.5/7X,12HSLENDER BODY6DB20 18
     2,1X,4F10.5/7X,9HCROSSFLOW,4X,4F10.5/7X,9HEMPENNAGE,4X,5F10.5/7X,   6DB20 19
     3 63(1H-)/7X,5HTOTAL,8X,5F10.5)                                     6DB20 20
  703 FORMAT(/5X,31HLOAD AND VELOCITY DISTRIBUTIONS/14X,5HX, FT,6X,3HX/L6DB20 21
     1,5X,6HDCN/DX,4X,6HDCY/DX,5X,4HU/VS,6X,4HV/VS,6X,4HW/VS)            6DB20 22
  704 FORMAT(10X,9F10.5)                                                 6DB20 23
  705 FORMAT(/5X,68HLOCATION OF STORE IN DISPENSER COORDINATE SYSTEM, DI6DB20 24
     1MENSIONS IN FEET/17X,26HRELATIVE TO DISPENSER NOSE,9X,28HRELATIVE 6DB20 25
     2TO INITIAL POSITION/20X,2HXF,8X,2HYF,8X,2HZF,11X,6HDEL XF,4X,6HDEL6DB20 26
     3 YF,4X,6HDEL ZF)                                                   6DB20 27
  706 FORMAT(/5X,82HTRANSLATIONAL VELOCITIES AND ACCELERATIONS OF STORE 6DB20 28
     1IN DISPENSER COORDINATE SYSTEM/10X,28HRELATIVE TO DISPENSER MOTION6DB20 29
     2/15X,3HDXF,7X,3HDYF,7X,3HDZF,6X,4HD2XF,6X,4HD2YF,6X,4HD2ZF)        6DB20 30
  707 FORMAT(/5X,75HROTATIONAL VELOCITIES AND ACCELERATIONS OF STORE IN 6DB20 31
     1STORE COORDINATE SYSTEM/16X,1HP,9X,1HQ,9X,1HR,7X,4HPDOT,6X,4HQDOT,6DB20 32
     26X,4HRDOT)                                                         6DB20 33
  708 FORMAT(/5X,92HSTORE ANGULAR ORIENTATION IN DISPENSER COORDINATE SY6DB20 34
     1STEM AND RATES OF CHANGE OF THESE ANGLES/10X,56HANGLES IN DEGREES,6DB20 35
     2 RATES OF CHANGE IN RADIANS PER SECOND/15X,3HPSI,6X,5HTHETA,6X,3HP6DB20 36
     3HI,6X,4HDPSI,5X,6HDTHETA,5X,4HDPHI)                                6DB20 37
  709 FORMAT(10X,4HNOSE,1X,3F10.5,5X,3F10.5)                             6DB20 38
  710 FORMAT(10X,4HXMOM,1X,3F10.5,5X,3F10.5)                             6DB20 39
  711 FORMAT(10X,4HBASE,1X,3F10.5,5X,3F10.5)                             6DB20 40

  601 FORMAT(/20X,29HCROSSFLOW-DRAG COEFFICIENT IS,F10.5)

  603 FORMAT(///5X,100(1H.)//6X,C5,2H =,F9.5)

       RTD = 57.29578

      IF (IPLT.EQ.1.AND.NPTS.EQ.1) THEN

  10  WRITE (4,401)
  401 FORMAT(1X,'ENTER NAME OF FILE:    - .PLT')
      READ (4,402) FPLOT
  402 FORMAT(A16)
      OPEN(8,FILE=FPLOT,ERR=20,STATUS='NEW',RECL=80,ACCESS='DIRECT')
      WRITE (8,801)
```

```
      ...RMAT(2X,'TIME (SEC)',3X,'THETA (DEG)',3X,'DTHETA (RPS)',4X,
     1       'Q (FPS)',9X,'CLM',/)
      GO TO 30

20    WRITE (4,403) FPLOT
403   FORMAT(/1X,'FILE NAME "',A12,'" IS IN USE,...TRY AGAIN')
      GO TO 10

      ENDIF

C       OUTPUT INDEPENDENT VARIABLE
C
30    IF (NV.EQ.0) THEN
        WRITE (6,701) TIME
      ELSE
        VALUE = VAR(NV)
        IF (NV.EQ.7) VNAME = 'XSMC'
        IF (NV.EQ.8) VNAME = 'YSMC'
        IF (NV.EQ.9) VNAME = 'ZSMC'
        IF (NV.GE.10) VALUE = VAR(NV)*RTD
        IF (NV.EQ.10) VNAME = 'PSI'
        IF (NV.EQ.11) VNAME = 'THETA'
        IF (NV.EQ.12) VNAME = 'PHI'
        WRITE (6,603) VNAME,VALUE
        IF
C
C       OUTPUT FORCES AND MOMENTS
C
      CNT=CNBY+CNSB+CNCF+CNEM
      CYT=CYBY+CYSB+CYCF+CYEM
      CLMT=CLMBY+CLMSB+CLMCF+CLMEM
      CLNT=CLNBY+CLNSB+CLNCF+CLNEM
      CLLT=CLLEM
      WRITE (6,702) CNBY,CYBY,CLMBY,CLNBY,CNSB,CYSB,CLMSB,CLNSB,CNCF,
     1CYCF,CLMCF,CLNCF,CNEM,CYEM,CLMEM,CLNEM,CLLEM,
     1                        CNT,CYT,CLMT,CLNT,CLLT
        ITE (6,601) CDC

      IF (ILVD.EQ.0) GO TO 40
C
C       OUTPUT LOAD AND VELOCITY DISTRIBUTIONS
C
      WRITE (6,703)
      DO 1 J=2,NHSEG,2
      K=J/2
      XL=EXST(J)/ESTLGC
    1 WRITE (6,704) EXST(J),XL,CNX(K),CYX(K),UT(J),VT(J),WT(J)

C       CALCULATE AND OUTPUT STORE LOCATION (NOSE, MOMENT CENTER,
C           AND BASE) IN FUSELAGE COORDINATE SYSTEM
C
40    XXX=XMOM
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XNOSE=VAR(7)+XI
      YNOSE=VAR(8)+ETA
       OSE=VAR(9)+ZETA
        =XNOSE-XNOSEI
        =YNOSE-YNOSEI
        =ZNOSE-ZNOSEI
        =-(SLTHC(NEJECT)-XMOM)
        L STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XBASE=VAR(7)+XI
      YBASE=VAR(8)+ETA
      ZBASE=VAR(9)+ZETA
      DXB=XBASE-XBASEI
      DYB=YBASE-YBASEI
      DZB=ZBASE-ZBASEI
      DXCG=VAR(7)-XCGI
      DYCG=VAR(8)-YCGI
```

```
      DZCG=VAR(9)-ZCGI
      WRITE (6,705)
      WRITE (6,709) XNOSE,YNOSE,ZNOSE,DXN,DYN,DZN
      WRITE (6,710) VAR(7),VAR(8),VAR(9),DXCG,DYCG,DZCG
      WRITE (6,711) XBASE,YBASE,ZBASE,DXB,DYB,DZB
      IF (NV.GE.7) GOTO 50
C
C     OUTPUT STORE VELOCITIES
C
      WRITE (6,706)
      WRITE (6,704) VAR(1),VAR(2),VAR(3),DVAR(1),DVAR(2),DVAR(3)
C
C     OUTPUT STORE ACCELERATIONS
C
      WRITE (6,707)
      WRITE (6,704) VAR(4),VAR(5),VAR(6),DVAR(4),DVAR(5),DVAR(6)
C
C     OUTPUT STORE ANGULAR ORIENTATION
C
      CONTINUE
      PSI = VAR(10)*RTD
      THA = VAR(11)*RTD
      PHI = VAR(12)*RTD
      WRITE (6,708)
      WRITE (6,704) PSI,THA,PHI,DVAR(10),DVAR(11),DVAR(12)

      IF (IPLT.EQ.1)  WRITE (8,802) TIME,THA,DVAR(11),VAR(5),CLMT
  802 FORMAT(2X,F9.5,4X,F10.5,5X,E10.4,4X,E10.4,3X,F10.5)

      RETURN
      END

C*****************************************************************
      SUBROUTINE SHAPE (X,NS,XE,C,R,DRDX,ISH)

C     SUBROUTINE TO CALCULATE SHAPE
C
      DIMENSION XE(ISH),C(ISH,7)
      DO 1 K=1,NS
      XL=XE(K)
      J=K
      IF (X.LE.XL) GO TO 2
    1 CONTINUE
    2 R=C(J,1)+X*C(J,5)+X*X*C(J,6)

      ARG=X*X*C(J,2)+X*C(J,3)+C(J,4)
      DRDX=C(J,5)+2.0*X*C(J,6)
      IF (ARG.LE.0.0) RETURN
      R=R+SQRT(ARG)*C(J,7)
      DRDX=DRDX+(2.0*X*C(J,2)+C(J,3))/(2.0*SQRT(ARG))*C(J,7)
      RETURN
      END

C*****************************************************************
      SUBROUTINE SIMSON (N,F,DX,SUM)

C     SIMPSON RULE SUBROUTINE
      DIMENSION F(101)
      SUM=F(1)+F(N)
      DO 1 I=2,N,2
    1 SUM=SUM+4.0*F(I)
      M=N-2
      DO 2 I=3,M,2
    2 SUM=SUM+2.0*F(I)
      SUM=DX*SUM/3.0
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE SOROUT (N,X,S)

C     SUBROUTINE TO OUTPUT SOURCE DISTRIBUTION
C
      DIMENSION X(1),S(1)
  700 FORMAT(/15X,3HX/L,2X,8(1PE12.5))
  701 FORMAT(15X,5HQ*(K),8(1PE12.5))
      NA=N/8
      NB=8*NA
         (NB.LT.N) NA=NA+1
         K=1,NA
        8*(K-1)+1
      NC=NB+7
      IF (NC.GT.N ) NC=N
      WRITE (6,700) (X(L),L=NB,NC)
    1 WRITE (6,701) (S(L),L=NB,NC)
      RETURN
      END

C*********************************************************************
      SUBROUTINE SRCINPUT (NCASE,IBOD,NBODS,NSORC,BXL,BSOR,NBPOLY,
     *                     BXEND,BCOEF,ISR)

      CHARACTER*16 SRCF,DSRCF,SSRCF,SNSRCF

      DIMENSION  BXL(101),BSOR(101),BXEND(ISR),BCOEF(ISR,7)
      SAVE SRCF,DSRCF,SSRCF,SNSRCF
C
  101 FORMAT(7X,I2)
  102 FORMAT(10X,7F10.0)
  103 FORMAT(/7X,I4)
  104 FORMAT(/8X,6E12.4)
  105 FORMAT(8X,6E12.4)

  401 FORMAT(/1X,'ENTER NAME OF DISPENSER SOURCE FILE:  ( - .SRC)')
  402 FORMAT(/1X,'ENTER NAME OF SUBMISSILE SOURCE FILE:  ( - .SRC)')
  403 FORMAT(/1X,'ENTER NAME OF SUBMISSILE NO.',I3,' SOURCE FILE')
  404 FORMAT(A16)
  405 FORMAT(/1X,'USE PREVIOUS DISPENSER SOURCE FILE:  ',A12,


     *        '?  YES=1, NO=0')
  406 FORMAT(/1X,'USE PREVIOUS SUBMISSILE SOURCE FILE:  ',A12,
     *        '?  YES=1, NO=0')
  407 FORMAT(/1X,'USE PREVIOUS NAME OF SUBMISSILE NO.',I3,
     *        'SOURCE FILE',A12,'?  YES=1, NO=0')
  408 FORMAT(/1X,'FILE "',A12,'" NOT AVAILABLE....TRY AGAIN'/)
C
      IF(NCASE.GE.1) THEN
        IF(IBOD.LT.1) THEN
          WRITE (4,405) DSRCF
          READ(4,*) IPSF
          IF (IPSF.EQ.1) THEN
            SRCF=DSRCF
            GO TO 16
          END IF
          GO TO 14
        ELSE
          IF(NBODS.EQ.1) THEN
            WRITE(4,406) SSRCF
            READ(4,*) IPSF
            IF (IPSF.EQ.1) THEN
              SRCF=SSRCF
              GO TO 16
            END IF
            GO TO 14
          END IF
```

```
          IF (NBODS.GT.1) THEN
            WRITE(4,407) IBOD,SNSRCF
            READ(4,*) IPSF
            IF (IPSF.EQ.1) THEN
              SRCF=SNSRCF
              GO TO 16
            END IF
            GO TO 14
          END IF
        END IF
      END IF
   14 IF (IBOD.LT.1) THEN
        WRITE (4,401)
       ELSE
        IF (NBODS.EQ.1) WRITE (4,402)
        IF (NBODS.GT.1) WRITE (4,403) IBOD
      ENDIF
   12 READ (4,404) SRCF
       IF (IBOD.LT.1) THEN
         DSRCF=SRCF
       ELSE
         IF (NBODS.EQ.1) THEN
           SSRCF=SRCF
         END IF
         IF (NBODS.GT.1) THEN
           SNSRCF=SRCF
         END IF
       END IF
   16 OPEN (1,FILE= SRCF,ERR=18,STATUS='OLD',FORM= 'FORMATTED',
     *      ACCESS='DIRECT')
      GO TO 22

   18 WRITE(4,408) SRCF
      GO TO 12
C

   22 READ (1,101) NBPOLY
      READ (1,102) (BXEND(J), J=1,NBPOLY)
      DO 5 J=1,NBPOLY
    5 READ (1,102) (BCOEF(J,K), K=1,7)

      READ (1,103)  NSORC
      NAA=NSORC/6                                                       SORB1165
      NBB=6*NAA                                                         SORB1166
      IF (NBB.LT.NSORC) NAA=NAA+1                                       SORB1167
      DO 10 J=1,NAA
      NB=6*(J-1)+1                                                      SORB1169
      NC=NB+5                                                           SORB1170
      IF (NC.GT.NSORC) NC=NSORC                                         SORB1171
      READ (1,104) (BXL(N), N=NB,NC)
   10 READ (1,105) (BSOR(N), N=NB,NC)
      CLOSE(1)
      RETURN
      END
C*********************************************************************
      SUBROUTINE STRIO(NCASE)
C
C     SUBROUTINE TO INPUT AND OUTPUT STORE DATA                         6DB26   3
C
      DIMENSION  XEND(7),COEF(7,7),DUMX(101),DUMQ(101),
     1          SLTHI(10)

      COMMON /CONTROL/ INP,MODE,NEJECT,NFU,NPTS,NSTRS,NV
      COMMON /INDEX/ NCW,MSW,M,IMAX,IP(3),NCP(3),MSP(3),MP(3),MP1(3),   MPYLN222
     1              MMP(3),KMAX(3),NCWS,MS,NCPS(3),MPS(3)               MPYLN223
```

```
      COMMON /CONSTS/ DTR,RAD,RTD                                       6DB26 14
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF
      COMMON /STRDATA/ SLTHC(10),SRMAX(10),XSMC(10),YSMC(10),ZSMC(10),
     1                SIBCR(10),PHI(10),PSI(10),NSPOLY,SXEND(7),SIC(10),
     2                SCOEF(7,7),NSSOR(10),SSOR(101,10),SXL(101,10)
C
  701 FORMAT(8F10.0)                                                    6DB26 17
  702 FORMAT(///11X,16HSTORE INPUT DATA)                                6DB26 18
  703 FORMAT(/40X,38HMAXIMUM      MOMENT REFERENCE LOCATION,4X,9HINCIDEN6DB26 19
     1CE/21X,14HSTORE   LENGTH,5X,6HRADIUS,9X,21HBEHIND DISPENSER NOSE, 6DB26 20
     2 8X,5HANGLE/23X,2HNO,6X,2HFT,9X,2HFT,8X,5HX, FT,6X,5HY, FT,6X,    6DB26 21
     3  5HZ, FT,6X,3HDEG)                                               6DB26 22
  704 FORMAT(20X,I5,6F11.5)                                             6DB26 23
  705 FORMAT(//15X,32HSOURCE DISTRIBUTION FOR STORE NO,I3)              6DB26 24
  708 FORMAT(/45X,34HINCOMPRESSIBLE SOURCE DISTRIBUTION)                6DB26 27
C
      WRITE (6,702)                                                     6DB26 34
      WRITE(6,703)                                                      6DB26 35
      DO 2 J=1,NSTRS                                                    6DB26 28
      IF (INP.EQ.2)  READ (5,701) SLTHC(J),SRMAX(J),XSMC(J),YSMC(J),    6DB26 29
     1                           ZSMC(J),SIC(J),PHI(J),PSI(J)           6DB26 30
      SLTHI(J)=SLTHC(J)/BETA                                            6DB26 31
      ...R(J) = SIC(J)*DTR
      ... (6,704) J,SLTHC(J),SRMAX(J),XSMC(J),YSMC(J),ZSMC(J),SIC(J)    6DB26 37

      .. 3 J=1,NSTRS                                                    6DB26 42

      CALL SRCINPUT (NCASE,J,NSTRS,MSOR,DUMX,DUMQ,NPOLY,XEND,COEF,7)

      WRITE (6,705) J                                                   6DB26 46
      WRITE(6,708)                                                      6DB26 47
      CALL SOROUT (MSOR,DUMX,DUMQ)                                      6DB26 48

      IF (J.EQ.NEJECT) THEN
        DO 4 K=1,15
        IF (XEND(K).LE.1) THEN
          SXEND(K)=XEND(K)
          DO 8 L=1,7
    8     SCOEF(K,L)=COEF(K,L)
        ELSE
          GO TO 6
        ENDIF
    4   CONTINUE
    6   NSPOLY=K-1
      ENDIF
C
      DA = SLTHI(J)                                                     6DB26 51
      DB=DA*DA                                                          6DB26 52
      NSSOR(J) = MSOR                                                   6DB26 53
      DO 5 L=1,MSOR                                                     6DB26 54
      SXL(L,J) = -DA*DUMX(L)                                            6DB26 55
      ..R(L,J) = DB*DUMQ(L)                                             6DB26 56
      ..TINUE                                                           6DB26 59
      ..TURN                                                            6DB26 79
      ..D                                                               6DB26 80

C***********************************************************************
      SUBROUTINE STTOIN (X,Y,Z,XI,ETA,ZETA,DC)                          6DB27  1

C     SUBROUTINE TO TRANSFORM FROM STORE TO INERTIAL SYSTEM             6DB27  2
C                                                                       6DB27  3
      DIMENSION DC(3,3)                                                 6DB27  4
      XI=X*DC(1,1)+Y*DC(1,2)+Z*DC(1,3)                                  6DB27  5
      ETA=X*DC(2,1)+Y*DC(2,2)+Z*DC(2,3)                                 6DB27  6
      ZETA=X*DC(3,1)+Y*DC(3,2)+Z*DC(3,3)                                6DB27  7
      RETURN                                                            6DB27  8
      END                                                               6DB27  9
```

```
C***********************************************************************
      SUBROUTINE TABLE3 (IXTRP,W,WT,X,XT,NX,NPX,Y,YT,NY,NPY,Z,ZT,NZ,
     *                   NPZ)
C
C     TABLE LOOK-UP ROUTINE FOR 3 INDEPENDENT VARIABLES.
C     IXTRP = 1 EXTRAPOLATE IF NECESSARY.
C     IXTRP = 2 EXTRAPOLATION NOT ALLOWED. TAKE THE LAST(OR FIRST) POINT.
C     W     = ANSWER,(DEPENDENT VARIABLE CORRESPONDING TO INPUTS  X,Y,Z)
C     WT    = TABLE OF DEPENDENT VARIBALE CORRESPONDING TO XT,YT,ZT
C             WT(I,J,K)  INCREMENT SUBSCRIPTS LEFT TO RIGHT WHEN LOADING
C     X     = THE ARGUMENT OR INDEPENDENT VARIABLE  X
C     XT    = TABLE OF INDEP.  X  VALUES (MUST BE IN INCREASING ORDER)
C     NX    = NUMBER OF POINTS IN  XT
C     NPX   = NUMBER OF POINTS TO USE FOR  X  INTERPOLATION
C     Y     = THE ARGUMENT OR INDEPENDENT VARIABLE  Y
C     YT    = TABLE OF INDEP.  Y  VALUES (MUST BE IN INCREASING ORDER)
C     NY    = NUMBER OF POINTS IN  YT
C     NPY   = NUMBER OF POINTS TO USE FOR  Y  INTERPOLATION
C     Z     = THE ARGUMENT OR INDEPENDENT VARIABLE  Z
C     ZT    = TABLE OF INDEP.  Z  VALUES (MUST BE IN INCREASING ORDER)
C     NZ    = NUMBER OF POINTS IN  ZT
C     NPZ   = NUMBER OF POINTS TO USE FOR  Z  INTERPOLATION
C
      DIMENSION XT(NX),YT(NY),ZT(NZ),WT(NX,NY,NZ), B(20),A(10)
      CALL LIMIT (Z,ZT,NZ,NPZ,MINZ,MAXZ)
      CALL LIMIT (Y,YT,NY,NPY,MINY,MAXY)
      CALL LIMIT (X,XT,NX,NPX,MINX,MAXX)
      DO 41 K=MINZ,MAXZ


      DO 42 J=MINY,MAXY
   42 CALL INTERP (IXTRP,2,B(J),WT(1,J,K),X,XT,NX,NPX,MINX,MAXX)
   41 CALL INTERP (IXTRP,2,A(K),B,Y,YT,NY,NPY,MINY,MAXY)
      CALL INTERP (IXTRP,2,W,A,Z,ZT,NZ,NPZ,MINZ,MAXZ)
      RETURN

C***********************************************************
      SUBROUTINE VELFLD (XB,YB,ZB,UTU,VTV,WTW)                          6DB31   1
C
C     SUBROUTINE TO CALCULATE PERTURBATION VELOCITIES                   6DB31   2
C
      DIMENSION  SIBIR(10),CSIBIR(10),SSIBIR(10),XBSOI(10),YBSO(10),
     1           ZBSO(10)

      COMMON /CONTROL/ INP,MODE,NEJECT,NFU,NPTS,NSTRS,NV
      COMMON /INDEX/ NCW,MSW,M,IMAX,IP(3),NCP(3),MSP(3),MP(3),MP1(3),   MPYLN274
     1             MMP(3),KMAX(3),NCWS,MS,NCPS(3),MPS(3)                MPYLN275
      COMMON /CONSTS/ DTR,RAD,RTD                                       6DB31  11
      COMMON /FUSDATA/ FLTHI,FRMAX,NFSOR,FXL(101),FSOR(101),NFPOLY,     6DB3I  14
     1                FXEND(15),FCOEF(15,7)
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                6DB31  16
      COMMON /CFORCE/ DC(3,3),DX,UT(81),VAR(12),VT(81),WT(81),          6DB31  17
     1               ERAD(81),EDRDX(81),VSTORE,XMOM,ESTRMX,CNBY,CYBY,   6DB31  18
     2   CLMBY,CLNBY,DELX,CNX(40),CYX(40),CNSB,CYSB,CLMSB,CLNSB,CNCF,   6DB31  19
     3CYCF,CLMCF,CLNCF                                                  6DB31  20
      COMMON /FLOW/ ALFACR,BETA,BETASQ,FMACH,GAMF,RHO,SALFFI,VINF
      COMMON /STRDATA/ SLTHC(10),SRMAX(10),XSMC(10),YSMC(10),ZSMC(10),
     1                SIBCR(10),PHI(10),PSI(10),NSPOLY,SXEND(7),SIC(10),
     2                SCOEF(7,7),NSSOR(10),SSOR(101,10),SXL(101,10)

      IF (NFU.EQ.0) GO TO 10
C                                                                       6DB31  23
C     CALCULATE INCOMPRESSIBLE FUSELAGE INDUCED VELOCITIES              6DB31  24
C                                                                       6DB31  25
      CALL VELOC (XB,YB,ZB,FXL,FSOR,NFSOR,UF,VF,WF,FRMAX)               6DB31  27
      CALL BESKIN (XB,YB,ZB,VF,WF)                                      6DB31  28
      UTU=UTU+UF                                                        6DB31  31
      VTV=VTV+VF                                                        6DB31  32
      WTW=WTW+WF                                                        6DB31  33
```

```
                                                                       6DB31 64
C       CALCULATE VELOCITIES INDUCED BY OTHER STORES  .                6DB31 65
C                                                                      6DB31 66
   10 IF (NSTRS.LE.1) GO TO 50                                         6DB31 67
      DO 41 J=1,NSTRS                                                  6DB31 68
      IF (J.EQ.NEJECT) GO TO 41                                        6DB31 69
C                                                                      6DB31 70
C     LOCATE POINT IN STORE COORDINATE SYSTEM                          6DB31 71
C                                                                      6DB31 72
      XBSOI(J) = (-XSMC(J)+XMOM*COS(SIBCR(J)))/BETA
      YBSO(J) = YSMC(J)
      ZBSO(J) = ZSMC(J)-XMOM*SIN(SIBCR(J))
      SIBIR(J) = ATAN(BETA*TAN(SIBCR(J)))
      CSIBIR(J) = COS(SIBIR(J))
      SSIBIR(J) = SIN(SIBIR(J))

      XS=(XB-XBSOI(J))*CSIBIR(J)-(ZB-ZBSO(J))*SSIBIR(J)                6DB31 73
      YS=YB-YBSO(J)                                                    6DB31 74
      ZS=(XB-XBSOI(J))*SSIBIR(J)+(ZB-ZBSO(J))*CSIBIR(J)                6DB31 75
C                                                                      6DB31 76
C     CALCULATE INDUCED VELOCITIES                                     6DB31 77
C                                                                      6DB31 78
      CALL VELOC(XS,YS,ZS,SXL(1,J),SSOR(1,J),NSSOR(J),UF,VF,WF,SRMAX(J) 6DB31 79


     1)                                                                6DB31 80
      UTU  =UTU  +UF*CSIBIR(J)+WF*SSIBIR(J)                            6DB31 81
      VTV  =VTV  +VF                                                   6DB31 82
      WTW  =WTW  +WF*CSIBIR(J)-UF*SSIBIR(J)                            6DB31 83
   41 CONTINUE                                                         6DB31 84
C                                                                      6DB31 86
      TRANSFORM VELOCITIES TO COMPRESSIBLE SPACE AND RESOLVE INTO STORE 6DB31 87
      COORDINATE SYSTEM                                                6DB31 88
                                                                       6DB31 89
   50 UTT = UTU/BETASQ                                                 6DB31 90
      VTT=VTV  /BETA                                                   6DB31 91
      WTT=WTW  /BETA                                                   6DB31 92
      CALL INTOST (UTT,VTT,WTT,UTU,VTV,WTW,DC)                         6DB31 93
      RETURN                                                           6DB31 94
      END                                                              6DB31 95

C**********************************************************************
      SUBROUTINE VELOC (X,Y,Z,XS,SS,NS,U,V,W,RMAX)                     6DB32  1

C     SUBROUTINE TO CALCULATE PERTURBATION VELOCITIES AT A FIELD POINT 6DB32  2
C       OF AN AXISYMMETRIC BODY REPRESENTED BY POINT SOURCES AND SINKS 6DB32  3
C                                                                      6DB32  4
      DIMENSION XS(1),SS(1)                                            6DB32  5
      RS = Y*Y + Z*Z                                                   6DB32  6
      R = SQRT(RS)                                                     6DB32  7
      U=0.0                                                            6DB32  8
      IF (NS.EQ.0) GO TO 7                                             6DB32  9
      VR=0.0                                                           6DB32 10
      IF (X.GT.0.0) GO TO 5                                            6DB32 11
      IF (X.LT.XS(NS)) GO TO 5                                         6DB32 12
      IF (R.LE.0.0) GO TO 7                                            6DB32 13
      IF (R-RMAX) 4,4,5                                                6DB32 14
    4 PSI=0.5*RS                                                       6DB32 15
      DO 6 N=1,NS                                                      6DB32 16
      DX=X-XS(N)                                                       6DB32 17
      D=SQRT(DX*DX+RS)                                                 6DB32 18
    6 PSI=PSI-SS(N)*(1.0-DX/D)                                         6DB32 19
      IF (PSI) 7,7,5                                                   6DB32 20
    7 [illegible]=0.0                                                  6DB32 21
      [illegible].0                                                    6DB32 22
      [illegible]URN                                                   6DB32 23
      [illegible] 1 N=1,NS                                             6DB32 24
      DX=X-XS(N)                                                       6DB32 25
      F = DX*DX + RS                                                   6DB32 26
      D = SS(N)/(F*SQRT(F))                                            6DB32 27
      U = U + DX*D                                                     6DB32 28
```

```
    1 VR = VR + R*D                                          6DB32 29
      IF (Y) 3,2,3                                           6DB32 30
    2 W=VR                                                   6DB32 31
      V=0.0                                                  6DB32 32
      IF (Z.LT.0.0) W=-W                                     6DB32 33
      RETURN                                                 6DB32 34
    3 PHI=ATAN(ABS(Z/Y))                                     6DB32 35
      V=VR*COS(PHI)                                          6DB32 36
      W=VR*SIN(PHI)                                          6DB32 37
      IF (Z.LT.0.0) W=-W                                     6DB32 38
      IF (Y.LT.0.0) V=-V                                     6DB32 39
      RETURN                                                 6DB32 40
      END                                                    6DB32 41
$BEND
```

APPENDIX B.6

SUBSONIC TRAJECTORY EXECUTIVE FILE
NEARSUB.CSS

```
***** LOADS AND RUNS SUBSONIC STORE PROGRAM *****
*
*
 LOAD @0
 AS 4,CON:
 $IFNULL @1
   AS 5,NULL:
   XAL @0.OPT,IN,132
   AS 6,@0.OPT
 $WRITE OUTPUT FILE: @0.OPT
 $ELSE
   $IFX @1.INP
     AS 5,@1.INP
   $ELSE
     AS 5,NULL:
   $ENDC
   XAL @1.OPT,IN,132
   AS 6,@1.OPT
 $WRITE OUTPUT FILE: @1.OPT
 $ENDC
 START
 $EXIT
```

NOTE: If a formatted input file is to be used, its name must be typed in right after the CSS filename.

APPENDIX B.7

INPUT DATA, SEQUENCE, FORMAT, AND DEFINITIONS FOR FORMATTED FILE OPERATION

Item 1 NCARDS FORMAT(I5)

NCARDS is the number of cards of alphanumeric data used to describe the problem.

Item 2 HEAD FORMAT(20A4)

HEAD is NCARDS lines of alphanumeric data selected by the user to describe the problem.

Item 3 NFU, NSTRS, NEJECT FORMAT(3I5)

- NFU Dispenser missile present? 0 = No, 1 = Yes.
- NSTRS Number of submissiles present, $0 \leq$ NSTRS ≤ 10.
- NEJECT Number of submissiles ejected, can be 1 to 10. Data for this submissile must be read at the NEJECT position in the DO LOOP of subroutine STRIO.

Item 4 FMACH, RHO, VINF, GAMF, ALFAC FORMAT(5F10.0)

- FMACH Free stream Mach number.
- RHO Static air density at flight test conditions, slugs per cubic foot.
- VINF Free-stream velocity, feet per second.
- GAMF Dispenser missile flight path angle, degrees.
- ALFAC Dispenser missile angle of attack, degrees.

Item 5 FLTHC, FRMAX FORMAT (2F10.0)

- FLTHC Actual length (no wake) of dispenser missile fuselage, feet.
- FRMAX Dispenser missile maximum radius, feet.

Item 6 SLTHC(J), SRMAX(J), XSNC(J), YSN(J), ZSN(J), SIC(J), PHI(J), PSI(J) FORMAT (8F10.0)

- SLTHC(J) Jth submissile length, feet.
- SRMAX(J) Jth submissile maximum radius, feet.
- XSMC(J) Longitudinal distance from dispenser missile nose to a point XMOM (Item 9) aft of Jth submissile nose, feet; positive aft of dispenser missile nose.
- YSMC(J) Lateral distance from dispenser missile center line to a point on the Jth submissile center line XMOM aft of submissile nose feet; positive to right.
- ZSMC(J) Vertical distance from dispenser missile center line to a point on the Jth submissile center line XMOM aft of submissile nose, feet; positive below.

NOTE: For data base purposes it was desirable to input the submissile moment reference center position. Calculations require the nose position. The distance from the nose to moment reference center of the separating submissile was selected as the location criteria for all submissiles.

SIC(J) Jth submissile pitch incidence angle measured from dispenser missile center line, degrees; positive nose up.

PHI(J) Jth submissile roll angle with respect to the dispenser missile degrees; positive clockwise about X-axis.

PSI(J) Jth submissile yaw angle with respect to the dispenser missile, degrees; positive nose to the right.

NOTE: There must be a line of data consisting of the values of Item 6 for each of the submissiles, in sequence.

The following Items 7 through 12 pertain to the separating submissile.

Item 7 NSEG, NSEGXO, NGAM, NROLL, NEMP, NDAMP, NV FORMAT(7I5)

NSEG	Number of equal length segments the submissile body is divided into for the force calculation; NSEG $\leq$ 40.
NSEGXO	Number of body segments to the flow separation location.
NGAM	Trajectory to simulate wind tunnel captive store trajectory? 0 = No, 1 = Yes (Does not affect the single variable sweep).
NROLL	Rolling moment to be calculated? 0 = No, 1 = Yes.
NEMP	Empennage present? 0 = No, 1 = Yes.
NDAMP	Damping to be included in force calculation? 0 = No, 1 = Yes.
NV	Variable number which controls the type of calculation to be performed. NV = 0, separation trajectory. NV = 7, submissile performs X-sweep. NV = 9, submissile performs Z-sweep. NV = 11, submissile performs -sweep.

Item 8 SMASS, FIXX, FIYY, FIZZ, FIYZ, FIXZ, FIXY FORMAT(7F10.0)

SMASS	Submissile mass, slugs
FIXX	I_{XX} moment of inertia, slug-ft^2.
FIYY	I_{YY} moment of inertia, slug-ft^2.
FIZZ	I_{ZZ} moment of inertia, slug-ft^2.
FIYZ	I_{YZ} product of inertia, slug-ft^2.
FIXZ	I_{XZ} product of inertia, slug-ft^2.
FIXY	I_{XY} product of inertia, slug-ft^2.

Item 9 XMOM, XBAR, YBAR, ZBAR FORMAT(4F10.0)

XMOM	Location along submissile axis about which the pitching and yawing moments are to be taken, negative behind nose, feet; same point about which moments of inertia are taken.

XBAR X location of submissile center of gravity measured from moment center, feet; positive forward.

YBAR Y location of submissile center of gravity measured from submissile axis, feet; positive to the right.

ZBAR Z location of submissile center of gravity measured from submissile axis, feet; positive below.

Item 10 CA FORMAT(F10.0)

CA Submissile axial force coefficient relative to the dispenser missile axial force coefficient; reference area is submissile maximum cross-sectional area.

The next two items are input only if the submissile has an empennage (NEMP = 1).

Item 11 IPLNR, MSF FORMAT(2I5)

IPLNR = 0, cruciform empennage.

IPLNR = 1, planar empennage.

MSF = number of spanwise control points on each fin, must be odd and $5 \leq$ MSF ≤ 11.

Item 12 XTAIL, RADAV, FINSS, FINROL, CROOT, CTIP, SWPLE FORMAT(7F10.0)

XTAIL X location where the fin leading edge intersects the body surface, measured from submissile nose, feet; negative number.

RADAV Average submissile body radius in empennage region, feet; positive number.

FINSS Tail fin semispan, measured from submissile longitudinal axis, feet; positive number.

FINROL Initial fin orientation, degrees, $0 \leq$ FINROL ≤ 90; FINROL = 0 if fins vertical and horizontal.

CROOT Tail fin root chord, feet; positive number.

CTIP Tail fin tip chord, feet; positive number.

SWPLE Tail fin leading edge sweep angle, degrees; positive for swept back leading edge.

Item 13 VXZERO, VYZERO, VZZERO, VAR(4), VAR(5), VAR(6) FORMAT(6F10.0)

VXZERO Submissile initial longitudinal velocity, ft/s; positive forward.

VYZERO Submissile initial lateral velocity, ft/s; positive to the right.

VZZERO Submissile initial vertical velocity, ft/s; positive downward.

VAR(4) Submissile initial roll rate, P, radians/s.

VAR(5) Submissile initial pitch rate, Q, radians/s.

VAR(6) Submissile initial yaw rate, R, radians/s.

Item 14 DTIME, TIMEI, TIMEF FORMAT (3F10.0)

DTIME Integration interval, seconds.

TIMEI Initial time, seconds.

TIMEF Final time, seconds.

If a trajectory is to be run either from T = 0 or from a restart time, NV = 0, and Item 14 is input. For a trajectory restart, the last time step in the previous calculations output file is used to obtain the initial conditions. A trajectory restart requires that Item 14B be input.

Item 14B VAR[12 values] FORMAT(8F10.0)

This item consists of two cards with VAR(1) through VAR(8) on the first card and VAR(9) through VAR(12) on the second. The table below gives the notation used to identify VAR(1) through VAR(12) on the trajectory program output.

Program Notation	Output Notation
VAR(1)	DXF, ft/s
VAR(2)	DYF, ft/s
VAR(3)	DZF, ft/s
VAR(4)	P, radians/s
VAR(5)	Z, radians/s
VAR(6)	R, radians/s
VAR(7)	XF of XMOM, ft
VAR(8)	YF of XMOM, ft
VAR(9)	ZF of XMOM, ft
VAR(10)	PSI, degrees
VAR(11)	THETA, degrees
VAR(12)	PHI, degrees

Multiple cases may be run by using sequential sets of data in the input file.

DTIME is the integration interval to be used in the integration subroutine. The required interval depends, at least to a certain extent, upon the scale of the problem. For aircraft/store sized vehicles, Reference 6 reports that values from 0.05 to 0.10 second yield satisfactory results. It is suggested that cases be run with different time intervals to verify the validity of the time interval.

If a single variable sweep is to be run, NV = 0, and Item 15 must be input; however, Item 14 is omitted.

Item 15 FV,VI FORMAT(2F10.0)

FV is the final value of the variable to be varied;
VI is the increment by which the variable is varied

The variable may be swept from a small value to a larger value using a positive increment or it may be vice versa.

APPENDIX B.8

FORMATTED INPUT FILE FOR MULTIPLE SUBMISSILE CASE

```
    9
      THIS INPUT FILE IS SET UP TO CHECK OUT THE MULTIPLE
      SUBMISSILE OPTION OF THE FINAL VERSION OF THE SUBSONIC
      TRAJECTORY PROGRAM.  ALL OF THE CONFIGURATIONS ARE
      FROM THE DATA BASE.  THE DISPENSER MISSILE HAS COVERED
      SUBMISSILE BAYS.  SUBMISSILE #1 IS THE TWO CALIBER
      OGIVE NOSE FROM 'ASUBMIS'.  SUBMISSILE #2 IS THE ONE-
      HALF CALIBER OGIVE NOSE FROM 'ASUBMIS'.  SUBMISSILE #3
      IS THE HIGH FINENESS RATIO CONFIGURATION FROM 'BSUBMIS'.
      LET THE #2 SUBMISSILE BE THE ONE EJECTED.
    1     3     2
    0.8       0.001760    849.7        0.0        0.0
 2.968333     0.156250
0.466083      0.038833     0.733      -0.170      0.2200        0.0
0.466083      0.038833     0.733       0.0        0.2200        0.0
0.933333      0.033333     0.733      +0.170      0.2200        0.0
   36    36      0      0      0      0     9
   0.1         0.001       0.01       0.01        0.0        0.0        0.0
  0.23300       0.0         0.0        0.0
    2.0
    0.0         0.0         0.0        0.0        0.0        0.0
   1.2          0.2
```

APPENDIX C

SUPERSONIC TRAJECTORY PROGRAM INPUT DATA DEFINITIONS AND PROGRAM LISTINGS

APPENDIX C.1

SUPERSONIC TRAJECTORY PROGRAM NAMELIST INPUT DATA FILE DEFINITIONS

HEAD1 HEAD2 HEAD3	Three lines of alphanumeric text of the user's choice to define the problem, configuration, etc.
GAMF	Dispenser missile flight path angle, degrees.
FMACH	Free stream Mach number.
RHO	Static air density at flight altitude or wind tunnel test conditions, slugs per cubic foot.
VINF	Free Stream velocity, feet per second.
NFU	Dispenser missile present? NFU = 0, No; NFU = 1, Yes.
NSTRS	Number of stores; present version limited to NSTRS = 1.
FLTHC	Dispenser missile fuselage length, feet.
FRMAX	Dispenser missile fuselage maximum radius, feet.
NFPOLY	Number of polynomials (body segments) specifying the dispenser missile fuselage shape; $1 \leq$ NFPOLY ≤ 15.
FXEND(J)	X/ℓ values of end points of the polynomials specifying dispenser missile fuselage shape, NFPOLY values.
NFSOR	Number of line sources and line doublets to be used over the dispenser missile fuselage length; NFSOR ≤ 100.
SLTHC	Submissile fuselage length, feet.
SRMAX	Submissile fuselage maximum radius, feet.
PHI	Submissile roll angle relative to inertial system, degrees.
PSI	Submissile yaw angle relative to inertial system, degrees.
MSOR	Number of line sources and line doublets to be used to model the submissile volume and angle of attack effects; MSOR < 100.
NSPOLY	Number of polynomials (body segments) specifying the submissile shape; $1 \leq$ NSPOLY ≤ 7.

SXEND(J) X/ℓ values of the end points of the body segments defining the complete submissile shape; NSPOLY values.

NSEG Number of equal length segments the submissile body is divided into for the force calculation; NSEG $\leq$ 40.

NSEGXO Number of submissile body segments to the flow separation location for non-linear cross flow force calculations.

NGAM Trajectory to simulate wind tunnel captive-store trajectory? NGAM = 0, No; NGAM = 1, Yes. Inactive for single parameter sweep calculations.

NROLL Rolling moment to be calculated? NROLL = 0, No; NROLL = 1, Yes.

NEMP Empennage present? NEMP = 0, No; NEMP = 1, Yes.

NDAMP Damping to be included in force calculation? NDAMP = 0, No; NDAMP = 1, Yes.

SMASS Submissile mass, slugs (arbitrary for single variable sweep).

FIXX I_{XX} moment of inertia, slug-ft^2 (arbitrary for single variable sweep).

FIYY I_{YY} moment of inertia, slug-ft^2 (arbitrary for single variable sweep).

FIZZ I_{ZZ} moment of inertia, slug-ft^2 (arbitrary for single variable sweep).

FIYZ I_{YZ} product of inertia, slug-ft^2 (arbitrary for single variable sweep).

FIXZ I_{XZ} product of inertia, slug-ft^2 (arbitrary for single variable sweep).

FIXY I_{XY} product of inertia, slug-ft^2 (arbitrary for single variable sweep).

XMOM Location along submissile axis about which the pitching and yawing moments are to be taken, negative behind nose, feet; same point about which moments of inertia are taken.

XBAR X location of submissile C.G., measured from the moment center, feet; positive forward.

YBAR Y location of submissile C.G., measured from the submissile axis, feet; positive to the right.

ZBAR Z location of submissile C.G., measured from the submissile axis, feet; positive below.

CA Submissile axial force coefficient; reference area is submissile maximum cross-sectional area.

IPLNR Cruciform empennage, IPLNR = 0
Planar empennage, IPLNR = 1

MSF Number of spanwise control points on each fin for force calculation; must be odd and $5 \leq \text{MSF} \leq 11$.

XTAIL X location of tail fin leading edge-body radius juncture from submissile nose, feet; negative number.

RADAV Average submissile body radius in empennage region, feet; positive number.

FINSS Tail fin semispan, measured from submissile body longitudinal axis, feet; positive number.

FINROL Initial fin orientation, degrees, $0 \leq \text{FINROL} \leq 90$; FINROL = 0 if fins are vertical and horizontal.

CROOT Tail fin root chord, feet; positive number.

CTIP Tail fin tip chord, feet; positive number.

SWPLE Tail fin leading edge sweep angle, degrees. Measured from normal to submissile longitudinal axis; sweep back is positive.

IAFBOD Variable to indicate if there is a submissile fuselage aft of fin trailing edge for fin lift carry over on the body calculation; IAFBOD = 0, no afterbody; IAFBOD = 1, long afterbody.

NFXTYP(J) Type of shape for each body segment of the dispenser missile; NFPOLY values
NFXTYP(I) = 1; segment is an ogive (circular arc).
NFXTYP(I) = 2; segment is a cone or conical frustum.
NFXTYP(I) = 3; segment is a cylinder.

NOTE: A set of geometric data must be input for each body segment and the data for each segment depends upon the segment's type.

For circular arc segments; NFXTYP = 1.

FXO(I) Axial location of the center of the circular arc, feet; measured from the dispenser missile nose, positive number.

FRO(I) Vertical location of the center of the circular arc; feet, measured from dispenser missile axis, negative for a convex section, positive for a concave section.

FCRO(I) Radius of the circular arc, feet, positive number.

NOTE: The arc equations are set up to generate a body profile in the upper (+ r) plane. If a convex section has a radius or curvature (R_O) less than the body radius (r) then FRO(I) would be a positive number.

For conical frustum segments; NFXTYP = 2.

FXI(I) Axial location of the segment start, feet; positive number measured from the dispenser missile nose.

FRI(I) Radius of the dispenser missile at the start of the frustum, feet; positive number.

FXF(I) Axial location of the segment end, feet; positive number measured from the dispenser missile nose.

FRF(I) Radius of the dispenser missile at the end of the frustum, feet; positive number.

For cylindrical segments; NFXTYP = 3.

FRCYL(I) Radius of the dispenser missile along the length of the cylindrical segment, feet; positive number.

NSXTYP(I) Type of shape for each body segment of the submissile; NSPOLY values
NSXTYP(I) = 1, segment is an ogive (circular arc).
NSXTYP(I) = 2, segment is a cone or conical frustum.
NSXTYP(I) = 3, segment is a cylinder.

For circular arc segments, NSXTYP = 1.

SXO(I) Axial location of the center of the circular arc, feet; measured from the submissile nose, positive number.

SRO(I) Vertical location of the center of the circular arc, feet; measured from submissile axis, negative for a convex section, positive for a concave section.

SCRO(I) Radius of the circular arc, feet; positive number.

For conical frustum segments; NSXTYP = 2.

SXI(I) Axial location of the segment start, feet; positive number measured from the submissile nose.

SRI(I) Radius of the submissile at the start of the frustum, feet; positive number.

SXF(I) Axial location of the segment end, feet; positive number measured from the submissile nose.

SRF(I) Radius of the submissile at the end of the frustum, feet; positive number.

For cylindrical segments; NSXTYP = 3.

SRCYL(I) Radius of the submissile along the length of the cylindrical segment, feet; positive number.

VXZERO Submissile initial longitudinal velocity with respect to the dispenser missile, feet per second; positive forward.

VYZERO Submissile initial lateral velocity with respect to the dispenser missile, feet per second; positive to right.

VZZERO Submissile initial vertical velocity with respect to the dispenser missile, feet per second; positive down.

VAR(4) Submissile initial roll rate, radians/sec.

VAR(5) Submissile initial pitch rate, radians/sec.

VAR(6) Submissile initial yaw rate, radians/sec.

VAR VAR(1) to VAR(12), values from a trajectory calculation at time t to restart the trajectory calculation. The following table gives the notation used to identify VAR(1) through VAR(12) on the trajectory output.

PROGRAM NOTATION	OUTPUT NOTATION
VAR(1)	DXF, ft/sec.
VAR(2)	DYF, ft/sec.
VAR(3)	DZF, ft/sec.
VAR(4)	P, radians/sec.
VAR(5)	Q, radians/sec.
VAR(6)	R, radians/sec.
VAR(7)	XF of XMOM, ft
VAR(8)	YF of XMOM, ft
VAR(9)	ZF of XMOM, ft
VAR(10)	PSI, degrees
VAR(11)	THETA, degrees
VAR(12)	PHI, degrees

APPENDIX C.2

SUPERSONIC TRAJECTORY PROGRAM SCREEN CUES AND INTERACTIVE INPUT DATA

CUE: Enter Run Mode
1 is an Alpha Sweep
2 is a Computed Trajectory
3 is a User Selected Variable Sweep
4 is an X Sweep
5 is a Z Sweep
READ (4,*) Mode

CUE: Enter Dispenser Missile Angle of Attack, Degrees
READ (4,*) ALFAC

CUE: Enter X-Location of Store Moment Center
Measured from the Dispenser Nose, Feet
READ (4,*) XSMC

NOTE: XSMC is positive aft of dispenser missile nose.

CUE: Enter Y-Location of Store Moment Center
Measured from the Dispenser Centerline, Feet
READ (4,*) YSMC

CUE: Enter Z-Location of Store Moment Center
Measured from the Dispenser Centerline, Feet
READ (4,*) ZSMC

NOTE: ZSMC is positive below the dispenser missile centerline.

CUE: Enter Store Angle of Incidence
Relative to Dispenser Centerline, Degrees
READ (4,*) SIC

NOTE: SIC is positive, submissile nose up towards dispenser missile centerline.

If the calculation is an angle of attack sweep, the following two cues appear:

CUE: Enter Final Value of Store Angle of Incidence,
Degrees
READ (4,*) FV

CUE: Enter Incremental Value of Store
Angle of Incidence
READ (4,*) VI

If the calculation is a physical separation trajectory, the following three cues appear:

```
CUE:  Enter Initial Time, Seconds
      READ (4,*) TIMEI

CUE:  Enter Final Time, Seconds
      READ (4,*) TIMEF

CUE:  Enter Time Increment, Seconds
      READ (4,*) DTIME
```

If the calculation is a longitudinal (X) sweep, the following two cues appear:

```
CUE:  Enter Final X-Location of Store Moment Center,
      Feet
      READ (4,*) FV

CUE:  Enter Incremental Value of Store X-Translation
      Feet
      READ (4,*) VI
```

If the calculation is a vertical (Z) sweep, the following two cues appear:

```
CUE:  Enter Final Z-Location of Store Moment Center,
      Feet
      READ (4,*) FV

CUE:  Enter Incremental Value of Store Z-Translation,
      Feet
      READ (4,*) VI
```

If a user selected sweep is chosen, the following three cues appear. They define program sweep capabilities and request the appropriate data.

```
CUE:  NUMBER     VARIABLE
       8         YSMC.....Y-Location of Store Moment Center
      10         PSI.....Yaw Angle of Store
      12         PHI.....Roll Angle of Store
      Enter Number of Selected Variable
      READ (4,*) NV

CUE:  Enter Final Value of Variable
      READ (4,*) FV

CUE   Enter Variable Increment
      READ (4,*) VI
```

APPENDIX C.3

EXECUTIVE FILE FOR THE SUPERSONIC TRAJECTORY PROGRAM
NEARSUP.CSS

```
***** LOADS AND RUNS SUPERSONIC STORE PROGRAM *****
*
*
  LOAD @0
  AS 4,CON:
  $IFNULL @1
    AS 5,NULL:
    XAL @0.OPT,IN,132
    AS 6,@0.OPT
  $WRITE OUTPUT FILE: @0.OPT
  $ELSE
    $IFX @1.INP
      AS 5,@1.INP
    $ELSE
      AS 5,NULL:
    $ENDC
    XAL @1.OPT,IN,132
    AS 6,@1.OPT
  $WRITE OUTPUT FILE: @1.OPT
  $ENDC
  START
  $EXIT
```

APPENDIX C.4

Namelist Input File for Configuration N2D1
HIN2D1.NAM

```
HEAD1 = 'ANALYSIS FOR ASUBMIS DATA BASE CONFIGURATION N2D1',
HEAD2 = 'THE DISPENSER MISSILE HAS COVERED SUBMISSILE BAYS',
HEAD3 = 'THE SUBMISSILE HAS A TWO CALIBER OGIVE NOSE & NO FINS',
GAMF= 0.0,
FMACH = 1.20,
RHO = 0.0019909,
VINF = 1342.4,
NFU = 1,
NSTRS=1,
FLTHC = 2.968333,
FRMAX = 0.156667,
NFPOLY = 2,
FXEND(1) = 0.9375,
FXEND(2) = 2.968333,
NFXTYP(1) = 1,
FXO(1) = 0.9375,
FRO(1) = -2.7340,
FCRO(1) = 2.890667,
NFXTYP(2) = 3,
FRCYL(2) = 0.1566670,
NFSOR = 45,
SLTHC = 0.466083,
SRMAX = 0.0388330,
PHI = 0.0
PSI = 0.0
NSPOLY = 2,
SXEND(1) = 0.1533330,
SXEND(2) = 0.4660830,
NSXTYP(1) = 1,
SXO(1) = 0.1533330,
SRO(1) = -0.29125,
SCRO = 0.3300833,
NSXTYP(2) = 3,
SRCYL(2) = 0.0388330,
NSOR = 40,
NSEG = 36,
NSEGXO = 36,
NGAM=0,
NROLL = 1,
NEMP = 0,
NDAMP = 0,
SMASS = 1.0,
FIXX = 1.0,
FIYY = 1.0,
FIZZ = 1.0,
XMOM = -0.2330,
CA = 2.0,
NSF = 7,
XTAIL = 0.43275,
RADAV = 0.0388330,
FINSS = 0.0971670,
FINROL = 0.0,
CROOT = 0.033333,
CTIP = 0.033333,
SWPLE = 0.0,
IAFBOD = 0,
VZZERO = 0.0/
```

APPENDIX C.5

Namelist Input File for Test Case Dispenser with Eleven Body Segments

```
HEAD1 = 'ANALYSIS FOR ASUBMIS DATA BASE CONFIGURATION N1D2',
HEAD2 = 'THE DISPENSER MISSILE HAS THREE OPEN SUBMISSILE BAYS',
HEAD3 = 'THE OPEN BAYS ARE MODELED WITH A SERIES OF CIRCULAR ARCS',
GAMF= 0.0,
FMACH = 1.20,
RHO = 0.0019909,
VINF = 1342.4,
NFU = 1,
NSTHS=1,
FLTHC = 2.15,
FRMAX = 0.15625,
NFPOLY = 11,
FXEND(1) = 0.5000,
FXEND(2) = 0.6,
FXEND(3) = 0.75,
FXEND(4) = 0.985417,
FXEND(5) = 1.200000,
FXEND(6) = 1.325000,
FXEND(7) = 1.476083,
FXEND(8) = 1.597438,
FXEND(9) = 1.840146,
FXEND(10) = 1.961500,
FXEND(11) = 2.15,
NFXTYP(1) = 1,
FXO(1) = 0.9375,
FRO(1) = -2.734417,
FCRO(1) = 2.890667,
NFXTYP(2) = 2,
FXI(2) = 0.5,
FRI(2) = .122951,
FXF(2) = 0.600,
FRF(2) = 0.1264765,
NFXTYP(3) = 2,
FXI(3) = 0.6,
FRI(3) = 0.1264765,
FXF(3) = 0.75,
FRF(3) = 0.1318,
NFXTYP(4) = 1,
FXO(4) = 0.68888376,
FRO(4) = 1.86584002,
FCRO(4) = 1.73511671,
NFXTIP(5) = 2,
FXI(5) =0.9854170,
FRI(5) = 0.156250,
FXF(5) = 1.200000,
FRF(5) = 0.150000,
NFXTYP(6) =2,
FXI(6) = 1.200000,
FRI(6) = 0.150000,
FXF(6) = 1.325000,
FRF(6) = 0.14500,
NFXTYP(7) = 1,
FXO(7) = 1.348166,
FRO(7) = 0.957237,
FCRO(7) = 0.811139
NFXTYP(8) = 1,
FXO(8) = 1.476083,
FRO(8) = -1.251150,
FCRO(8) = 1.40700,
NFXTYP(9) = 1,
FXO(9) = 1.718792,
FRO(9) = 1.553161,
```

```
FCRU(9) = 1.407400,
NFXTYP(10) =1,
FXU(10) = 1.961500,
FRU(10) = -1.251143,
FCRU(10) = 1.407395,
NFXTYP(11) =3,
FRCYL(11) = 0.156250,
NFSOR = 95,
SLTHC = 0.466083,
SRMAX = 0.0388330,
PHI = 0.0,
PSI = 0.0,
NSPOLY = 2,
SXEND(1) = 0.0388330,
SXEND(2) = 0.4660830,
NSXTYP(1) = 1,
SXO(1) = 0.0388330,
SRO(1) = 0.0,
SCRO = 0.0388330,
NSXTYP(2) = 3,
SRCYL(2) = 0.0388330,
MSOR = 40,
NSEG = 36,
NSEGXU = 36,
NGAM=0,
NROLL = 1,
NEMP = 0,
NDAMP = 0,
SMASS = 1.0,
FIXX = 1.0,
FIYY = 1.0,
FIZZ = 1.0,
XMOM = -0.2330,
CA = 2.0,
MSF = 7,
XTAIL = 0.43275,
RADAV = 0.0388330,
FINSS = 0.0971670,
FINROL = 0.0,
CROOT = 0.033333,
CTIP = 0.033333,
SWPLE = 0.0,
IAFBOD = 0,
VZZERO = 0.0/
```

APPENDIX C.6

FORTRAN FILE FOR THE SUPERSONIC TRAJECTORY PROGRAM
NEARSUP.FTN

```
.BATCH
      PROGRAM SSTORE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)                SPA01   1
                                                                           SPA01   2
         GRAM TO CALCULATE SIX-DEGREE-OF-FREEDOM STORE TRAJECTORIES        SPA01   3
             AT SUPERSONIC SPEEDS                                          SPA01   4
                                                                           SPA01   5

      CHARACTER*16  NLFILE
      CHARACTER*80 HEAD1,HEAD2,HEAD3
C                                                                          SPA01   7
C     DIMENSION STATEMENTS                                                 SPA01   8
C                                                                          SPA01   9
      DIMENSION  BYY(50,7),FVN(10,11),HEAD(20)                             SPA01  11
C                                                                          SPA01  12
      COMMON STATEMENTS                                                    SPA01  13
                                                                           SPA01  14
      COMMON /CFORCE/ CLMBY,CLMCF,CLMSB,CLNBY,CLNCF,CLNSB,CNBY,CNCF,       SPA01  19
     1CNSB,CNX(40),CYBY,CYCF,CYSB,CYX(40),DC(3,3),DELX,DX,EDRDX(81),       SPA01  20
     2ERAD(81),ESTRMX,UT(81),VAR(12),VSTORE,VT(81),WT(81),XMOM             SPA01  21
      COMMON /CONTROL/ NFU,NV
      COMMON /CONSTS/ PI,DTOR,RTOD                                         SPA01  23
      COMMON /COUTPT/ DVAR(12),ESTLGC,EXST(81),TIME                        SPA01`24
      COMMON /EFORCE/ CNEM,CLMEM,CYEM,CLNEM,CLLEM                          SPA01  25
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,FINROL,MSF,IPLNR,CLALPH            SPA01  26
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ                       SPA01  28
      COMMON /FSGEOM/ FXEND(15),FCOEF(15,7),FLTHC,NFPOLY,NFXTYP(15),
                     FXO(15),FRO(15),FCRO(15),FXI(15),FRI(15),FXF(15),
                     FRF(15),FRCYL(15)
         ON /FSOR/ FXL(101),FSS(100),FDS(100),NFSOR
         ON /ICVEL/ UP,VP,WP,II,IF                                         SPA01  30
         ON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                    SPA01  31
         MON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,YBASEI,SPA01  34
     1ZBASEI                                                               SPA01  35
      COMMON /STGEOM/ SRMAX,SLTHC,XSMC,YSMC,ZSMC,SIC,SIBCR,PHI,PSI,        SPA01  40
     1                   MSOR,NSPOLY,SXEND(7),SCOEF(7,7),NSXTYP(7),        SPA01  41
     2               SXO(7),SRO(7),SCRO(7),SXI(7),SRI(7),SXF(7),
     3               SRF(7),SRCYL(7)
C
      NAMELIST /INPUT/ HEAD1,HEAD2,HEAD3,GAMF,FMACH,RHO,VINF,NFU,
     1                NSTRS,FLTHC,FRMAX,NFPOLY,FXEND,NFSOR,SLTHC,
     '                SRMAX,PHI,PSI,MSOR,NSPOLY,SXEND,NSEG,NSEGXO,NGAM,
                      NROLL,NEMP,NDAMP,SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXY,
                      FIXZ,XMOM,XBAR,YBAR,ZBAR,CA,IPLNR,MSF,XTAIL,
                      RADAV,FINSS,FINROL,CLALPH,VXZERO,VYZERO,VZZERO,
     6                VAR,NFXTYP,FXO,FRO,FCRO,FXI,FRI,FXF,FRF,FRCYL,
     7                NSXTYP,SXO,SRO,SCRO,SXI,SRI,SXF,SRF,SRCYL,
     8                CROOT,CTIP,SWPLE,IAFBOD

      FORMAT STATEMENTS                                                    SPA01  54
                                                                           SPA01  55
  701 FORMAT(10I5)                                                         SPA01  56
  702 FORMAT(//42X,55(1H*),/42X,55H* SUPERSONIC SIX-DEGREE-OF-FREEDOM TRSPA01  57
     1AJECTORY PROGRAM */42X,55(1H*),//)                                   SPA01  58
  703 FORMAT(20A4)                                                         SPA01  59
  704 FORMAT(10X,20A4)                                                     SPA01  60
  706 FORMAT(8F10.0)                                                       SPA01  62
```

```
707 FORMAT(///10X,17HFLIGHT CONDITIONS/15X,13HMACH NUMBER =,F5.2,       SPA01 63
   1/15X,26HFREE STREAM MASS DENSITY =,F10.7,21H SLUGS PER CUBIC FOOT,SPA01 64
   2/15X,22HFREE STREAM VELOCITY =,F8.2,16H FEET PER SECOND,          SPA01 65
   3/15X,19HFLIGHT PATH ANGLE =,F6.2,8H DEGREES)                      SPA01 66
7.. FORMAT(15X,27HDISPENSER ANGLE OF ATTACK =,F6.2,8H DEGREES)
7.. FORMAT(/20X,33HTHIS STORE HAS A PLANAR EMPENNAGE)                  SPA01 68
710 FORMAT(/20X,36HTHIS STORE HAS A CRUCIFORM EMPENNAGE)               SPA01 69
711 FORMAT(/24X,20HEMPENNAGE FORCES ACT,F9.5,17H FEET BEHIND NOSE,     SPA01 70
   1/24X,46HAVERAGE BODY RADIUS IN THE EMPENNAGE REGION IS,F9.5,      SPA01 71
   2 5H FEET)                                                         SPA01 72
714 FORMAT(/20X,27HAXIAL-FORCE COEFFICIENT IS ,F9.5)                   SPA01 76
717 FORMAT(//20X,12HSTORE MASS =,F10.5,6H SLUGS,                       SPA01 82
   1             /20X,45HMOMENTS AND PRODUCTS OF INERTIA, SLUG - SQ FSPA01 83
   2 //23X,5HIXX =,F10.5/23X,5HIYY =,F10.5/23X,5HIZZ =,F10.5/23X,5HIYZ SPA01 84
   3 =,F10.5/23X,5HIXZ =,F10.5/23X,5HIXY =,F10.5)                     SPA01 85
71  FORMAT(/20X,'STORE MOMENT CENTER IS',F9.5,' FEET BEHIND NOSE',/20XSPA01 86
   1,55HSTORE CENTER OF GRAVITY OFFSET FROM MOMENT CENTER, FEET/23X,6HSPA01 87
   2XBAR =,F9.5/23X,6HYBAR =,F9.5/23X,6HZBAR =,F9.5)                  SPA01 88
722 FORMAT(//20X,18HSEPARATION ASSUMED,F9.5,15H FEET FROM NOSE)        SPA01 92
729 FORMAT(24X,48HTAIL FIN SEMISPAN MEASURED FROM THE BODY AXIS IS,    SPA01 93
   1F9.5,5H FEET/24X,25HFINS ARE INITIALLY ROLLED,F6.2,41H DEGREES FROSPA01 94
   2M THE VERTICAL AND HORIZONTAL)                                    SPA01 95
735 FORMAT(24X,23HFIN LIFT-CURVE SLOPE IS,F9.5,11H PER RADIAN)         SPA01 96
754 FORMAT(/1X,18H*** NSTRS INPUT AS,I3,33H, PROGRAM IS LIMITED TO ONESPA01120
   1 STORE)                                                           SPA01121
C                                                                     SPA01122
      PI = 3.141593
      DTOR=0.0174532925
      RTOD=57.29577951
      ACCG = 32.174
                                                                      SPA01126
      NCASE = 0
      NOBOPT = 0

      READ (5,701,END=5) NCARDS
      WRITE(4,401)
401  FORMAT(1X,60(1H*)/1X,'INPUT SOURCE:   FORMATTED FILE')
      INP = 2
      GO TO 1000

 5   WRITE (4,402)
402  FORMAT(1X,60(1H*)/1X,'INPUT SOURCE:   TERMINAL/NAMELIST')
      INP = 1

C        INTERACTIVE DATA INPUT

 10  CALL ACTIVINP  (IPASS,MODE,ALFAC,DTIME,TIMEI,TIMEF,FV,VI)

      IF (NCASE.GE.1) THEN
        WRITE (4,403) NLFILE
403  FORMAT(/1X,'USE PREVIOUS NAMELIST FILE: ',A12,'?  YES=1,  NO=0')
        READ (4,*) IPNL
        IF (IPNL.EQ.1) GO TO 17
      ENDIF

 15  WRITE(4,405)
405  FORMAT(/1X,'ENTER NAMELIST FILE:  ( - .NAM)')
      READ (4,406) NLFILE
406  FORMAT(A16)

 17  OPEN(2,FILE= NLFILE,ERR=20,STATUS='OLD',FORM='FORMATTED',
    1     ACCESS='DIRECT')
      READ (2,INPUT)
        ..E (2)
        ..E(6,702)
        ..TE (6,601) HEAD1,HEAD2,HEAD3
 ..  ..RMAT(3(/10X,A80))
      GO TO 30
```

```
   20 WRITE (4,410) NLFILE
  410 FORMAT(/1X,'NAMELIST FILE "',A12,'" NOT AVAILABLE....TRY AGAIN')
      GO TO 15

C       FORMATTED HEADER

 1000 IF (NCASE.GE.1)  READ (5,701,END=2200) NCARDS
      WRITE(6,702)                                                      SPA01131
      DO 25 J=1,NCARDS                                                  SPA01132
      READ(5,703) HEAD                                                  SPA01133
   25 WRITE (6,704) HEAD                                                SPA01134

C       SPECIFY CONFIGURATION                                           SPA01146
C                                                                       SPA01147
      IF (INP.NE.1)  READ (5,701) NFU,NSTRS                             SPA01148
   30 IF (NSTRS.GE.2)  WRITE (4,754) NSTRS                              SPA01149
C                                                                       SPA01136
C       FLIGHT CONDITIONS                                               SPA01137
C                                                                       SPA01138
      IF (INP.NE.1)  READ (5,706) FMACH,RHO,VINF,GAMF,ALFAC             SPA01339
      WRITE (6,707) FMACH,RHO,VINF,GAMF                                 SPA01140
      IF (NFU.EQ.1)  WRITE (6,708) ALFAC
      BETASQ= (FMACH**2)-1.0                                            SPA01142
        A=SQRT(BETASQ)                                                  SPA01143
        ACR=ALFAC*DTOR                                                  SPA01144
                                                                        SPA01145
         (NFU.EQ.1) THEN

        INPUT FUSELAGE DATA                                             SPA01153

      CALL FUSEIO (INP,FRMAX)                                           SPA01157

      ENDIF
      IF (NSTRS .EQ. 0) GO TO 2100                                      SPA01211

        INPUT STORE DATA IF PRESENT                                     SPA01208

      CALL STORIO (INP)                                                 SPA01212

      LOCATE STORE IN FUSELAGE COORDINATE SYSTEM                        SPA01214

      PSIR = PSI*DTOR
      SSIBCR= SIN(SIBCR)                                                SPA01217
      CSIBCR= COS(SIBCR)                                                SPA01218
      SPSI = SIN(PSIR)

       PUT ADDITIONAL INFORMATION REQUIRED TO DESCRIBE EJECTED STORE    SPA01278
                                                                        SPA01279
         INP.EQ.1) GO TO 40
          (5,701) NSEG,NSEGXO,NGAM,NROLL,NEMP,NDAMP,NV                  SPA01281
          (5,706) SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXZ,FIXY                   SPA01282
          (5,706) XMOM,XBAR,YBAR,ZBAR                                   SPA01283
         TE (6,717) SMASS,FIXX,FIYY,FIZZ,FIYZ,FIXZ,FIXY                 SPA01285
      X OM=ABS(XMOM)                                                    SPA01287
      WRITE (6,718) XMOM,XBAR,YBAR,ZBAR
C                                                                       SPA01297
C     DETERMINE GEOMETRIC PARAMETERS DESCRIBING STORE                   SPA01298
C                                                                       SPA01299
      ESTLGC=SLTHC                                                      SPA01304
      ESTRMX=SRMAX                                                      SPA01305
      A  D= ESTLGC/ESTRMX
         F=PI*ESTRMX**2                                                 SPA01306
         L=2.0*ESTRMX                                                   SPA01307
```

```
      FSEG=NSEG                                                         SPA01308
      DELX=ESTLGC/FSEG                                                  SPA01309
      DX= 0.5*DELX                                                      SPA01310
      XST(1)=0.0                                                        SPA01311
      NHSEG=2*NSEG+1                                                    SPA01312
      DO 55 J=2,NHSEG                                                   SPA01313
   55 EXST(J)=EXST(J-1)+DX                                              SPA01314
      DO 56 J=2,NHSEG,2                                                 SPA01315
      XXX=EXST(J)/ESTLGC                                                SPA01316
      CALL SHAPE (XXX,NSPOLY,SXEND,SCDEF,RR,EDRDX(J),7)                  SPA013
   56 ERAD(J)=ESTLGC*RR                                                 SPA01318
      FSEGXO=NSEGXO                                                     SPA01319
        EP=FSEGXO*DELX                                                  SPA01320
        IE(6,722) XSEP                                                  SPA01321
         INP.NE.1)  READ (5,706) CA                                     SPA01322
         E (6,714) CA                                                   SPA01323
         GO=2*NSEGXO+1                                                  SPA01324
         M=0                                                            SPA01325
      IF (ABS(XBAR).GT.1.0E-05) NASYM=1                                 SPA01326
      IF (ABS(YBAR).GT.1.0E-05) NASYM=1                                 SPA01327
      IF (ABS(ZBAR).GT.1.0E-05) NASYM=1                                 SPA01328
      SAPG=SIN(ALFACR+GAMF*DTOR)                                        SPA01329
      CAPG=COS(ALFACR+GAMF*DTOR)                                        SPA01330
C                                                                       SPA01331
      INPUT EMPENNAGE DATA IF EMPENNAGE IS PRESENT                      SPA01332
                                                                        SPA01333
        M=0.0                                                           SPA01334
         M=0.0                                                          SPA01335
         =0.0                                                           SPA01336
        EM=0.0                                                          SPA01337
        EM=0.0                                                          SPA01338
      IF(NEMP.EQ.0) GO TO 57
      IF (INP.EQ.1) GO TO 50
      READ (5,701) IPLNR,MSF,IAFBOD                                     SPA01340
      READ (5,706) XTAIL,RADAV,FINSS,FINROL,CROOT,CTIP,SWPLE            SPA01341

   50 CALL FINCALC2  (FMACH,BETA,ESTRMX,FINSS,RADAV,CROOT,CTIP,SWPLE,
     +               IAFBOD,XTAIL,CLALPH)

      IF (IPLNR.EQ.1) WRITE (6,709)                                     SPA01342
      IF (IPLNR.EQ.0) WRITE (6,710)                                     SPA01343
      WRITE (6,711) XTAIL,RADAV                                         SPA01344
      WRITE (6,729) FINSS,FINROL                                        SPA01345
      WRITE (6,735) CLALPH                                              SPA01346
      XTAIL = -XTAIL

       ALL EMPINI                                                       SPA01347
                                                                        SPA01362
      INITIALIZE FOR TRAJECTORY CALCULATION                             SPA01363
C                                                                       SPA01364
   57 VAR(10) = PSIR                                                    SPA01366
      VAR(11) = SIBCR                                                   SPA01365
      VAR(12) = PHI*DTOR                                                SPA01367
      IF (INP.NE.1)  READ (5,706) VXZERO,VYZERO,VZZERO,                 SPA01368
     1              VAR(4),VAR(5),VAR(6)                                SPA01369
      VAR(1)=VZZERO*SSIBCR+VXZERO*CSIBCR                                SPA01370
      VAR(2)=VYZERO                                                     SPA01371
      VAR(3)=VZZERO*CSIBCR-VXZERO*SSIBCR                                SPA01372
      VAR(7) = -XSMC                                                    SPA01374
      VAR(8) = YSMC                                                     SPA01375
      VAR(9) = ZSMC                                                     SPA01376
      XNOSEI = -XSMC+XMOM*CSIBCR                                        SPA01377
      YNOSEI = YSMC+XMOM*SPSI                                           SPA01378
```

```
      ZNOSEI = ZSMC+XMOM*SSIBCR                                         SPA01379
      XCGI=VAR(7)                                                       SPA01380
      YCGI=VAR(8)                                                       SPA01381
      ZCGI=VAR(9)                                                       SPA01382
      XBASEI=XNOSEI-SLTHC*CSIBCR                                        SPA01383
      YBASEI = YNOSEI-SLTHC*SPSI                                        SPA01384
      ZBASEI=ZNOSEI+SLTHC*SSIBCR                                        SPA01385

      IF (NV.EQ.0) THEN
        IF (INP.EQ.2)  READ (5,706) DTIME,TIMEI,TIMEF                  SPA01386
        IF (TIMEI.GT.1.E-6) THEN
          IF (INP.EQ.2)  READ (5,706) VAR
          DO 61 J=10,12
 61       VAR(J) = DTOR*VAR(J)
        ENDIF
      ELSE
        DTIME = 0.0
        TIMEI = 0.0
        TIMEF = 0.0

        IF (INP.EQ.2)  READ (5,706) FV,VI
        IF (NV.EQ.7)  FV = -FV

        IF (NV.GE.10.AND.NV.LE.12) THEN
          FV = FV*DTOR
          VI = VI*DTOR
        ENDIF

        VI = ABS(VI)
        IF (VAR(NV).GT.FV)  VI = -VI
        TIMEF = 99.0
      ENDIF

      DDTIME = DTIME                                                    SPA01388
      TIME=TIMEI                                                        SPA01389
      NEQ=12                                                            SPA01387
      NDIFEQ = 1                                                        SPA01394
      CALL ADAMS (DTIME,DDTIME,VAR,DVAR,NEQ,NDIFEQ,TIME,NV,FV,VI)
      NOUT=1                                                            SPA01396
C                                                                       SPA01397
C     CALCULATE AERODYNAMIC FORCES AND MOMENTS                          SPA01398
C                                                                       SPA01399
      NPTS=0
   62 CALL SFORCE (FMACH,ALDD,CDC)                                      SPA01400
      IF (NEMP.EQ.1) CALL SEMFOR                                        SPA01401
      IF (NGAM.EQ.1) CALL DIRCOS(VAR,DC)                                SPA01402

      IF (NV.EQ.7.OR.NV.EQ.9.OR.NV.EQ.11) GO TO 90
C                                                                       SPA01403
C     CALCULATE ACCELERATIONS                                           SPA01404
C                                                                       SPA01405
C     CALCULATE COEFFICIENT MATRIX                                      SPA01406
C                                                                       SPA01407
      DO 70 J=1,6                                                       SPA01408
      DO 70 K=1,6                                                       SPA01409
   70 FVN(J,K)=0.0                                                      SPA01410
      FVN(1,1)=1.0                                                      SPA01411
      FVN(2,2)=1.0                                                      SPA01412
      FVN(3,3)=1.0                                                      SPA01413
      FVN(4,4)=FIXX                                                     SPA01414
      FVN(5,5)=FIYY                                                     SPA01415
      FVN(6,6)=FIZZ                                                     SPA01416
      FVN(4,5)=-FIXY                                                    SPA01417
```

```
      FVN(4,6)=-FIXZ                                                    SPA01418
      FVN(5,6)=-FIYZ                                                    SPA01419
      FVN(5,4)=-FIXY                                                    SPA01420
      FVN(6,4)=-FIXZ                                                    SPA01421
      FVN(6,5)=-FIYZ                                                    SPA01422
      IF (NASYM.EQ.0) GO TO 80                                          SPA01423
      FVN(1,4)=YBAR*DC(1,3)-ZBAR*DC(1,2)                                SPA01424
      FVN(1,5)=ZBAR*DC(1,1)-XBAR*DC(1,3)                                SPA01425
      FVN(1,6)=XBAR*DC(1,2)-YBAR*DC(1,1)                                SPA01426
      FVN(2,4)=YBAR*DC(2,3)-ZBAR*DC(2,2)                                SPA01427
      FVN(2,5)=ZBAR*DC(2,1)-XBAR*DC(2,3)                                SPA01428
      FVN(2,6)=XBAR*DC(2,2)-YBAR*DC(2,1)                                SPA01429
      FVN(3,4)=YBAR*DC(3,3)-ZBAR*DC(3,2)                                SPA01430
      FVN(3,5)=ZBAR*DC(3,1)-XBAR*DC(3,3)                                SPA01431
      FVN(3,6)=XBAR*DC(3,2)-YBAR*DC(3,1)                                SPA01432
      DO 71 J=4,6                                                       SPA01433
      DO 71 K=1,3                                                       SPA01434
   71 FVN(J,K)=SMASS*FVN(K,J)                                           SPA01435
C                                                                       SPA01436
C     CALCULATE RIGHT HAND SIDE                                         SPA01437
C                                                                       SPA01438
   80 GXX=-ACCG*(SAPG*DC(1,1)-CAPG*DC(3,1))                             SPA01439
        Y=-ACCG*(SAPG*DC(1,2)-CAPG*DC(3,2))                             SPA01440
         =-ACCG*(SAPG*DC(1,3)-CAPG*DC(3,3))                             SPA01441
         URE=0.5*RHO*VSTURE**2                                          SPA01442
         EF=QSTORE*SREF                                                 SPA01443
       SREFL=QSREF*REFL                                                 SPA01444
      CNORM=CNSB+CNBY+CNCF                                              SPA01445
      CSIDE=CYSB+CYBY+CYCF                                              SPA01446
      CROLL=0.0                                                         SPA01447
      CPITCH=CLMSB+CLMBY+CLMCF                                          SPA01448
      CYAW=CLNSB+CLNBY+CLNCF                                            SPA01449
      IF (NEMP.EQ.0) GO TO 81                                           SPA01450
      CNORM=CNORM+CNEM                                                  SPA01451
      CSIDE=CSIDE+CYEM                                                  SPA01452
      CROLL=CROLL+CLLEM                                                 SPA01453
       PITCH=CPITCH+CLMEM                                               SPA01454
       YAW=CYAW+CLNEM                                                   SPA01455
   81 CONTINUE                                                          SPA01456
      RONE=GXX-QSREF*CA/SMASS                                           SPA01460
      RTWO=GYY+QSREF*CSIDE/SMASS                                        SPA01462
      RTHR=GZZ-QSREF*CNORM/SMASS                                        SPA01463
      FVN(4,7)=QSREFL*CROLL                                             SPA01464
      FVN(5,7)=QSREFL*CPITCH                                            SPA01465
      FVN(6,7)=QSREFL*CYAW                                              SPA01466
      IF (NASYM.EQ.0) GO TO 85                                          SPA01467
        =RONE+XBAR*(VAR(5)**2+VAR(6)**2)-YBAR*VAR(4)*VAR(5)-ZBAR*VAR(4SPA01468
         (6)                                                            SPA01469
        RTWO-XBAR*VAR(4)*VAR(5)+YBAR*(VAR(4)**2+VAR(6)**2)-ZBAR*VAR(5SPA01470
         (6)                                                            SPA01471
        RTHR-XBAR*VAR(4)*VAR(6)-YBAR*VAR(5)*VAR(6)+ZBAR*(VAR(4)**2+VASPA01472
          **2)                                                          SPA01473
      FVN(4,7)=FVN(4,7)+SMASS*(GZZ*YBAR-GYY*ZBAR)                       SPA01480
      FVN(5,7)=FVN(5,7)+SMASS*(GXX*ZBAR-GZZ*XBAR)                       SPA01481
      FVN(6,7)=FVN(6,7)+SMASS*(GYY*XBAR-GXX*YBAR)                       SPA01482
   85 FVN(1,7)=RONE*DC(1,1)+RTWO*DC(1,2)+RTHR*DC(1,3)                   SPA01483
      FVN(2,7)=RONE*DC(2,1)+RTWO*DC(2,2)+RTHR*DC(2,3)                   SPA01484
      FVN(3,7)=RONE*DC(3,1)+RTWO*DC(3,2)+RTHR*DC(3,3)                   SPA01485
      FVN(4,7)=FVN(4,7)-VAR(6)*VAR(5)*(FIZZ-FIYY)+(VAR(5)**2-VAR(6)**2)*SPA01486
     1FIYZ+VAR(4)*(VAR(5)*FIXZ-VAR(6)*FIXY)                             SPA01487
       VN(5,7)=FVN(5,7)-VAR(6)*VAR(4)*(FIXX-FIZZ)+(VAR( )**2-VAR(4)**2)*SPA01488
          +VAR(5)*(VAR(6)*FIXY-VAR(4)*FIYZ)                             SPA01489
         6,7)=FVN(6,7)-VAR(4)*VAR(5)*(FIYY-FIXX)+(VAR(4)**2-VAR(5)**2)*SPA01490
```

```
     Y+VAR(6)*(VAR(4)*FIYZ-VAR(5)*FIXZ)

        LVE FOR ACCELERATIONS

      CALL INVERS (FVN,1,6,10,11)
      DO 86 J=1,6
   86 DVAR(J)=FVN(J,7)
      DO 87 J=7,9
   87 DVAR(J)=VAR(J-6)
      SPHI=SIN(VAR(12))
      CPHI=COS(VAR(12))
      DVAR(10)=(VAR(5)*SPHI+VAR(6)*CPHI)/COS(VAR(11))
      DVAR(11)=VAR(5)*CPHI-VAR(6)*SPHI
      DVAR(12)=VAR(4)+SIN(VAR(11))*DVAR(10)
C
C       OUTPUT TRAJECTORY DATA IF AT END OF INTEGRATION STEP
C
   90 IF (NOUT.EQ.0) GO TO 91
      IF (NPTS.NE.0) GO TO 210

      WRITE (4,411)
  411 FORMAT(/1X,'OUTPUT LOAD AND VELOCITY DISTRIBUTIONS?',/1X,'YES=1
     1NO=0')
      READ (4,*) ILVD
      IF (NV.EQ.0) THEN
        WRITE (4,412)
  412   FORMAT(/1X,'OUTPUT COLUMNED DATA FILE?  YES=1,  NO=0')
        READ (4,*) IPLT
      ENDIF
  210 NPTS= NPTS+1

      CALL SOUTPT (NPTS,IPLT,ILVD,CDC,THA,CNT,CYT,CLMT,CLNT,CLLT,PHI,
     *            PSI)
      NOUT=0

      IF (NV.NE.7.AND.NV.NE.9.AND.NV.NE.11) GO TO 180

C       CONSTRUCT DATA BASE OUTPUT ARRAY

      IF (NFU.EQ.1) THEN
        REFDIA = 2.*FRMAX
      ELSE
        WRITE (4,413)
  413 FORMAT(1X,'ENTER REFERENCE DIAMETER')
        READ (4,*) REFDIA
      ENDIF

      XS = -VAR(7)/REFDIA
      YS = VAR(8)/REFDIA
         = VAR(9)/REFDIA
         S = ALFAC+THA

         (NV.EQ.11) THEN
           MODE = 1
        BYY(NPTS,1) = ALPS
      ENDIF
      IF (NV.EQ.7) THEN
           MODE = 4
        BYY(NPTS,1) = XS
      ENDIF
      IF (NV.EQ.9) THEN
           MODE = 5
        BYY(NPTS,1) = ZS
```

```
      ENDIF

        BYY(NPTS,2) = CNT
        BYY(NPTS,3) = CYT
        BYY(NPTS,4) = CLMT
        BYY(NPTS,5) = CLNT
        BYY(NPTS,6) = CLLT
        BYY(NPTS,7) = CA

  180 IF (TIME+1.0E-05-TIMEF) 91,2000,2000                              SPA01511
C                                                                       SPA01512
C     CALL INTEGRATION ROUTINE                                          SPA01513
C                                                                       SPA01514
   91 IF (NV .EQ. 0)      GO TO 92
      NDIFEQ = 10
      CALL ADAMS (DTIME,DDTIME,VAR,DVAR,NEQ,NDIFEQ,TIME,NV,FV,VI)
      IF (NDIFEQ.EQ.1) GO TO 2000                                       SPA01516
      IF (NDIFEQ.GT.7) NOUT=1                                           SPA01517
      GO TO 62                                                          SPA01518

 2000 IF (NV.NE.7.AND.NV.NE.9.AND.NV.NE.11) GO TO 2100

      WRITE (4,440)
  440 FORMAT(/1X,'OUTPUT BINARY DATA BASE FILE?  YES=1, NO=0')
      READ (4,*) IFILE
      IF (IFILE.NE.1) GO TO 2100

C       OUTPUT DATA BASE FILE

      CALL DBASEOPT (NDBOPT,ALFAC,MODE,ALPS,NPTS,FMACH,PHI,PSI,XS,YS,
     +               ZS,BYY)
      NDBOPT = NDBOPT+1

 2100 NCASE = NCASE+1
      IF (INP.EQ.2) GO TO 1000
      WRITE (4,450)
  450 FORMAT(/1X,'DO YOU WISH TO MAKE ANOTHER RUN?  YES=1, NO=0')
      READ(4,*) IRUN
      IF (IRUN.EQ.1) GO TO 10

                                                                        SPA01519

C     *****************************************************************
      SUBROUTINE ACTIVINP (IPASS,MODE,ALFAC,DTIME,TIMEI,TIMEF,FV,VI)

C         SUBROUTINE INPUTS DATA FROM USER'S TERMINAL
C
      DIMENSION A(18)
      SAVE A

      COMMON /CONTROL/ NFU,NV
      COMMON /STGEOM/ SRMAX,SLTHC,XSMC,YSMC,ZSMC,SIC,SIBCR,PHI,PSI,
     1                MSOR,NSPOLY,SXEND(7),SCOEF(7,7),NSXTYP(7),
     2          SXO(7),SRO(7),SCRO(7),SXI(7),SRI(7),SXF(7),
     3          SRF(7),SRCYL(7)

      DTR = 0.01745329


      IF(IPASS.NE.0) GO TO 198
      IPASS=1
      DO 17 I=1,18
```

```
      A(I)=99.
17    CONTINUE
      WRITE(4,15)
   15 FORMAT(/1X,'ENTER RUN MODE',/5X,
     1'1 IS AN ALPHA SWEEP',/,5X,
       IS A COMPUTED TRAJECTORY',/5X,
       IS A USER SELECTED VARIABLE SWEEP',/5X,
       IS AN X SWEEP',/,5X,
       IS A Z SWEEP')
        (4,*) MODE
       ( )=MODE

      WRITE(4,32)
   32 FORMAT(/1X,'ENTER DISPENSER MISSILE ANGLE OF ATTACK, DEGREES')
      READ (4,*) ALFAC
      A(2)=ALFAC

      WRITE(4,20)
   20 FORMAT(/1X,'ENTER X-LOCATION OF STORE MOMENT CENTER',
     1 X,'MEASURED FROM THE DISPENSER NOSE, FEET')
        D (4,*) XSMC
        )=XSMC

      WRITE(4,25)
   25 FORMAT(1X,'ENTER Y-LOCATION OF STORE MOMENT CENTER',
     1/1X,'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      READ (4,*) YSMC
      A(6)=YSMC

      WRITE(4,30)
   30 FORMAT(1X,'ENTER Z-LOCATION OF STORE MOMENT CENTER',
     1/1X,'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      READ (4,*) ZSMC
      A(7)=ZSMC

      WRITE(4,35)
   35 FORMAT(1X,'ENTER STORE ANGLE OF INCIDENCE,',
     1/1X,'RELATIVE TO DISPENSER CENTERLINE, DEGREES')
      READ (4,*) SIC
      A(10)=SIC

      GO TO (40,50,80,60,70) MODE

   40 NV=11
      WRITE(4,42)
   42 FORMAT(/1X,'ENTER FINAL VALUE OF STORE ANGLE OF INCIDENCE,',
     1' DEGREES')
      READ (4,*) FV
      A(11)=FV
      WRITE(4,44)
   44 FORMAT(1X,'ENTER INCREMENTAL VALUE OF STORE',
     1/1X,'ANGLE OF INCIDENCE')
      READ (4,*) VI
      A(12)=VI
      GO TO 115

  50  NV = 0
      WRITE (4,52)
   52 FORMAT(/1X,'ENTER INITIAL TIME, SECONDS')
      READ (4,*) TIMEI
      A(13)=TIMEI
      WRITE(4,54)
   54 FORMAT(1X,'ENTER FINAL TIME, SECUNDS')
```

```
      READ (4,*) TIMEF
      A(14)=TIMEF
      WRITE(4,56)
   56 FORMAT(1X,'ENTER TIME INCREMENT, SECONDS')
      READ (4,*) DTIME
      A(15)=DTIME
      GO TO 115

   60 NV=7
      WRITE(4,62)
   62 FORMAT(/1X,'ENTER FINAL X-LOCATION OF STORE MOMENT CENTER,'
     1,' FEET')
      READ (4,*) FV
      A(4)=FV
      WRITE(4,64)
   64 FORMAT(1X,'ENTER INCREMENTAL VALUE OF STORE X-TRANSLATION,',
     1' FEET')
      READ (4,*) VI
      A(5)=VI
      GO TO 115

   70 NV=9
      WRITE(4,72)
   72 FORMAT(/1X,'ENTER FINAL Z-LOCATION OF STORE MOMENT CENTER,'
     1,' FEET')
      READ (4,*) FV
      A(8)=FV
      WRITE(4,74)
   74 FORMAT(1X,'ENTER INCREMENTAL VALUE OF STORE Z-TRANSLATION,',
     1' FEET')
      READ (4,*) VI
      A(9)=VI
      GO TO 115

   80 WRITE(4,82)
   82 FORMAT(/5X,'NUMBER',5X,'VARIABLE',/5X,6(1H-),5X,8(1H-),/8X,
        ',9X,'YSMC.....Y-LOCATION OF STORE MOMENT CENTER',/7X,
          ',9X,'PSI .....YAW ANGLE OF STORE',/7X,
          ',9X,'PHI .....ROLL ANGLE OF STORE',//1X,
         TER NUMBER OF SELECTED VARIABLE')
      READ (4,*) NV
      A(16)=NV
      WRITE(4,84)
   84 FORMAT(1X,'ENTER FINAL VALUE OF VARIABLE')
      READ (4,*) FV
      A(17)=FV
      WRITE(4,86)
   86 FORMAT(1X,'ENTER VARIABLE INCREMENT')
      READ (4,*) VI
      A(18)=VI
      GO TO 115

  THIS SECTION ALLOWS CHANGING OF CURRENT INTERACTIVE INPUT VALUES
C AFTER THE FIRST PASS
C
 198  WRITE(4,199)
 199  FORMAT(/1X,'ENTER 0 TO SEE MENU OF ALL INTERACTIVE INPUTS,'
     *,/1X,'OR 1 TO SEE IN SECTIONS'/)
      READ(4,*) ANS
      IF (ANS.EQ.1) GO TO 220
        TE(4,201) (A(I),I=1,18)
          AT(/T20'1    RUN MODE..................................'F10.2
             /T20'2    DISPENSER ANGLE OF ATTACK, DEG............'F10.2/
```

```
     *,       /T20'3   INITIAL X-LOCATION OF SUBMISSILE NOSE.....'F10.2
              /T20'4   FINAL X-LOCATION OF SUBMISSILE NOSE.......'F10.2
              /T20'5   INCREMENTAL X VALUE.......................'F10.2/
              /T20'6   Y-LOCATION OF SUBMISSILE NOSE.............'F10.2/
     *,       /T20'7   INITIAL Z-LOCATION OF SUBMISSILE NOSE.....'F10.2
     *,       /T20'8   FINAL Z-LOCATION OF SUBMISSILE NOSE.......'F10.2
     *,       /T20'9   INCREMENTAL Z VALUE.......................'F10.2/
     *,       /T20'10  INITIAL SUBMISSILE ANGLE OF INCIDENCE.....'F10.2
     *,       /T20'11  FINAL VALUE OF SUBMISSILE ANGLE ..........'F10.2
     *,       /T20'12  INCREMENTAL VALUE OF SUBMISSILE ANGLE.....'F10.2/
     *,       /T20'13  INITIAL TIME, SEC.........................'F10.2
     *,       /T20'14  FINAL TIME, SEC...........................'F10.2
     *,       /T20'15  TIME INCREMENT, SEC.......................'F10.2/
     *,       /T20'16  NUMBER OF VARIABLE TO BE VARIED...........'F10.2
     *,       /T20'17  FINAL VALUE OF VARIATION..................'F10.2
     *,       /T20'18  INCREMENT FOR THE VARIATION...............'F10.2)
 203  WRITE(4,302)
      READ(4,*) I,AI
      GO TO 400
C
C     RUN MODE
C
 300  IF(I.NE.1) GO TO 303
      WRITE(4,301)
 301  FORMAT(/T20'1 IS AN ALPHA SWEEP',
     *        /T20'2 IS A COMPUTED TRAJECTORY',
     *        /T20'3 IS A USER SELECTED VARIABLE SWEEP',
     *        /T20'4 IS AN X SWEEP',
     *        /T20'5 IS A Z SWEEP')
      GO TO 203
C
C     ALFAC
C
 303  IF(I.NE.2)GO TO 305
      WRITE(4,304)
 304  FORMAT(/T20'ENTER DISPENSER MISSILE ANGLE OF ATTACK, DEGREES')
      GO TO 203
C
C     XSMC(1)
C
 305  IF(I.NE.3)GO TO 307
      WRITE(4,306)
 306  FORMAT(/T20'ENTER INITIAL X-LOCATION OF SUBMISSILE NOSE,',
     *        /T20'MEASURED FROM THE DISPENSER NOSE, FEET')
      GO TO 203
C
C     XFIN
C
 307  IF(I.NE.4)GO TO 309
      WRITE(4,308)
 308  FORMAT(/T20'ENTER FINAL X-LOCATION OF SUBMISSILE NOSE, FEET')
      GO TO 203
C
C     XINC
C
 309  IF(I.NE.5)GO TO 311
      WRITE(4,310)
 310  FORMAT(/T20'ENTER INCREMENTAL VALUE OF SUBMISSILE,',
     *        /T20'X-TRANSLATION, FEET')
      GO TO 203
C
C     ZSMC(1)
C
```

```
 311  IF(I.NE.6)GO TO 313
      WRITE(4,312)
 312  FORMAT(/T20'ENTER Y-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      GO TO 203

      ZIC(1)

 313  IF(I.NE.7)GO TO 315
      WRITE(4,314)
 314  FORMAT(/T20'ENTER INITIAL Z-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
      GO TO 203

      ZFIN

 315  IF(I.NE.8)GO TO 317
      WRITE(4,316)
 316  FORMAT(/T20'ENTER FINAL Z-LOCATION OF SUBMISSILE NOSE,',
     1        /T20'MEASURED FROM THE DISPENSER CENTERLINE, FEET')
C
      ZINC

      IF(I.NE.9)GO TO 319
      WRITE(4,318)
 318  FORMAT(/T20'ENTER INCREMENTAL VALUE OF SUBMISSILE,',
     1        /T20'Z-TRANSLATION, FEET')
      GO TO 203
C
C     SIC(1)
C
 319  IF(I.NE.10)GO TO 321
      WRITE(4,320)
 320  FORMAT(/T20'ENTER INITIAL SUBMISSILE ANGLE OF INCIDENCE,',
     1        /T20'RELATIVE TO DISPENSER CENTERLINE, DEGREES')
      GO TO 203
C
C     ANGF
C
 321  IF(I.NE.11)GO TO 323
      WRITE(4,322)
 322  FORMAT(/T20'ENTER FINAL VALUE OF SUBMISSILE ANGLE OF INCIDENCE',
     1        /T20'RELATIVE TO DISPENSER CENTERLINE, DEGREES')
      GO TO 203
C
C     ANGI
C
 323  IF(I.NE.12)GO TO 325
      WRITE(4,324)
 324  FORMAT(/T20'ENTER INCREMENTAL VALUE OF SUBMISSILE',
     1        /T20'ANGLE OF INCIDENCE')
      GO TO 203

      TIMEI

      IF(I.NE.13)GO TO 327
      WRITE(4,326)
 326  FORMAT(/T20'ENTER INITIAL TIME, SECONDS')
      GO TO 203
C
      TIMEF
C
 327  IF(I.NE.14)GO TO 329
```

```
      WRITE(4,328)
 328  FORMAT(/T20'ENTER FINAL TIME, SECONDS')
      GO TO 203
C
C     DTIME
C
 329  IF(I.NE.15)GO TO 331
      WRITE(4,330)
 330  FORMAT(/T20'ENTER TIME INCREMENT, SECONDS')
      GO TO 203
C
C     XNV
C
 331  IF(I.NE.16)GO TO 333
      WRITE(4,332)
 332  FORMAT(/T20'ENTER NUMBER OF VARIABLE TO BE VARIED')
      GO TO 203
C
      VFIN

        I.NE.17)GO TO 335
        TE(4,334)
        MAT(/T20'ENTER FINAL VALUE OF VARIATION')
        TO 203

      VINC
C
 335  IF(I.NE.18)GO TO 203
      WRITE(4,336)
 336  FORMAT(/T20'ENTER INCREMENT FOR THE VARIATION')
      GO TO 203
C

  05  GO TO(500,510,520,530,540)MODE
      .V=11
      FV=A(11)
      VI=A(12)
      IF(A(11).EQ.99. .OR. A(12).EQ.99.)GOTO 550
      GO TO 110
510   IF(A(14).EQ.99. .OR. A(15).EQ.99.)GOTO 550
      GOTO 110
 520  FV=A(17)
      VI=A(18)
       (A(16).EQ.99. .OR. A(17).EQ.99. .OR. A(18).EQ.99.)
        TO 550
         110

        (4)
        (5)
        (A(4).EQ.99. .OR. A(5).EQ.99.)GOTO 550
      GO TO 110
 540  NV=9
      FV=A(8)
      VI=A(9)
      IF(A(7).EQ.99. .OR. A(8).EQ.99.)GO TO 550
      GO TO 110

      WRITE(1,410)
        MAT(//'** FINAL AND/OR INCREMENTAL VALUE
        SPECIFIED**'//)
        TO 203

 402  FORMAT(/T20'ENTER NUMBER, VALUE OR'
```

```
     1        /T20'ENTER 0,0 TO LIST CURRENT INPUTS OR'
     2        /T20'ENTER 0,1 TO RUN PROGRAM'
     3        /T20'ENTER NUMBER, 999, FOR MORE INFORMATION')
C
 400  IF(AI.EQ.999.)GO TO 300
      IF(I.EQ.0.AND.AI.EQ.0)GO TO 200
      IF(I.EQ.0.AND.AI.EQ.1)GO TO 105
      IF(I.EQ.1)MODE=AI
      IF(I.EQ.16)NV=AI
      A(I)=AI
      GO TO 203
C
C     ...E INPUT IN STEPS INSTEAD OF WITH LARGE MENU
C
 220  WRITE(4,221)
 221  FORMAT(/5X,'ENTER 0 TO CHANGE MODE, 1 TO GO ON'/)
      READ(4,*) ANS
      IF (ANS.EQ.1) GO TO 240
      WRITE(4,222)
 222  FORMAT (/1X'MODE CODES ---')
      WRITE(4,301)
      WRITE(4,*) ('ENTER NEW MODE?')
      READ(4,*) MODE
      A(1)=MODE
C
C     ... INITIAL CONDITIONS MENU
C
 240  WRITE(4,239)
 239  FORMAT (//1X'INITIAL CONDITIONS -- CURRENT STATUS'/)
      WRITE(4,241) A(2),A(3),A(6),A(7),A(10)
 241  FORMAT (/T20'1 = DISPENSER MISSILE ANGLE OF ATTACK, DEG...'F10.2
     1          /T20'2 = X-LOCATION OF SUBMISSILE NOSE............'F10.2
     2          /T20'3 = Y-LOCATION OF SUBMISSILE NOSE............'F10.2
     3          /T20'4 = Z-LOCATION OF SUBMISSILE NOSE............'F10.2
     4          /T20'5 = SUBMISSILE ANGLE OF INCIDENCE, DEG.......'F10.2/)
 244  WRITE(4, 245)
 245  FORMAT(/1X'ENTER  NUMBER,NEW VALUE  TO CHANGE AN INPUT, OR ',/1X,
     1          'ENTER  0,0  TO GO ON, OR ',/1X,
     2          'ENTER  0,1  TO SEE INITIAL VALUES AGAIN'/)
      READ(4,*) I,AI
      IF (I.EQ.0 .AND. AI.EQ.0) GO TO 250
      IF (I.EQ.0 .AND. AI.EQ.1) GO TO 240
      IF (I.EQ.1) THEN
        A(2)=AI
      ELSE IF (I.EQ.2) THEN
        A(3)=AI
      ELSE IF (I.EQ.3) THEN
        A(6)=AI
      ELSE IF (I.EQ.4) THEN
        A(7)=AI
      ELSE
        A(10)=AI
      ENDIF
      GO TO 244
C
C CHANGE INCREMENTS AND FINAL VALUES, IF DESIRED
C
 250  WRITE(4,251)
 251  FORMAT (/1X'ENTER 0 TO CHANGE INCREMENTS AND OTHER VALUES ',/1X
     *          'OR 1 TO RUN AGAIN'/)
      READ(4,*) ANS
      IF (ANS.EQ.1) GO TO 105
      GO TO (260,262,264,266,268) MODE
```

```
 260  WRITE(4, 261)
 261  FORMAT (5X'FOR MODE = 1'/)
      WRITE(4,322)
      READ(4,*) FV
      A(11)=FV
      WRITE(4,324)
      READ(4,*) VI
      A(12)=VI
      GO TO 105

 262  WRITE(4,263)
 263  FORMAT (5X'FOR MODE = 2'/)
      WRITE(4,326)
      READ(4,*) TIMEI
      A(13)=TIMEI
      WRITE(4,328)
      READ(4,*) TIMEF
      A(14)=TIMEF
      WRITE(4,330)
      READ(4,*) DTIME
      A(15)=DTIME
      GO TO 105
C
 264  WRITE(4,265)
 265  FORMAT (5X'FOR MODE = 3'/)
      WRITE(4,332)
      READ(4,*) NV
      A(16)=NV
      WRITE(4,334)
      READ(4,*) FV
      A(17)=FV
      WRITE(4,336)
      READ(4,*) VI
      A(18)=VI
      GO TO 105
C
 266  WRITE(4,267)
   7  RMAT (5X'FOR MODE = 4'/)
        IE(4,308)
        D(4,*) FV
       )=FV
      ITE(4,310)
      READ(4,*) VI
      A(5)=VI
      GO TO 105
C
 268  WRITE(4,269)
 269  FORMAT (5X'FOR MODE = 5'/)
      WRITE(4,316)
      READ(4,*) FV
      A(8)=FV
      WRITE(4,318)
      READ(4,*) VI
      A(9)=VI
      GO TO 105
C
C
 110  AGFAC=A(2)
      XSMC=A(3)
      YSMC=A(6)
      ZSMC=A(7)
       =A(10)
```

```
      TIMEI=A(13)
      TIMEF=A(14)
      DTIME=A(15)
 15   RETURN
      END

      **************************************************************
      SUBROUTINE ADAMS (H,DS,Y,DY,NEQ,NDIFEQ,S,NV,FV,VI)
C
C     ADAMS INTEGRATION ROUTINE
C      FIXED INTEGRATION INTERVAL
C
      DIMENSION Y(12),DY(12)
      DIMENSION Y1(12),DY3(12),Y5(12),E(12),Y3(12),DY1(12),Y2(12),PX(12)
     1,PF(12),DY2(12)
      GO TO (100,200,300,400,500,600,700,800,900,950),NDIFEQ
C
C     START BY RUNGE-KUTTA
C
  100 H=DS
      JB=1
      DO 101 I=1,NEQ
  101 Y1(I)=Y(I)
      SX1=S
  103 NDIFEQ=2
      RETURN
  200 DO 201 I=1,NEQ
       3(I)=DY(I)
         I)=Y(I)
          =H*DY(I)
          =0.5*TEMP+Y5(I)
          =TEMP
        .5*H+S
        IFEQ=3
      RETURN
  300 DO 301 I=1,NEQ
      TEMP=H*DY(I)
      Y(I)=0.5*TEMP+Y5(I)
  301 E(I)=E(I)+2.0*TEMP
      NDIFEQ=4
      RETURN
  400 DO 401 I=1,NEQ
        P=H*DY(I)
         )=Y5(I)+TEMP
          )=E(I)+2.0*TEMP
       =0.5*H+S
      NDIFEQ=5
      RETURN
  500 DO 501 I=1,NEQ
  501 Y(I)=(H*DY(I)+E(I))*0.16666667+Y5(I)
      GO TO (502,507,509,902),JB
  502 CONTINUE
  507 DO 508 I=1,NEQ
      DY1(I)=DY3(I)
  508 Y2(I)=Y(I)
      JB=3
      GO TO 103
  509 S=S-H
C
C     OUTPUT OF RUNGE-KUTTA
C
      IF (JB-4) 521,802,802
```

```
      NDIFEQ=8                                                          SPA02 55
      DO 522 I=1,NEQ                                                    SPA02 56
      PX(I)=Y(I)                                                        SPA02 57
      PF(I)=DY(I)                                                       SPA02 58
  522 Y(I)=Y2(I)                                                        SPA02 59
      RETURN                                                            SPA02 60
  800 DO 801 I=1,NEQ                                                    SPA02 61
      Y(I)=PX(I)                                                        SPA02 62
      DY(I)=PF(I)                                                       SPA02 63
  801 DY3(I)=DY3(I)                                                     SPA02 64
      ...=4                                                             SPA02 65
      S=S+H                                                             SPA02 66
  802 NDIFEQ=9                                                          SPA02 67
      RETURN                                                            SPA02 68
  900 IF (JB-5) 103,901,9901                                            SPA02 69
  901 CON=0.041666667*H                                                 SPA02 70
      JB=6                                                              SPA02 71
 9901 NDIFEQ=6                                                          SPA02 72
      GO TO 600                                                         SPA02 73
  902 JB=5                                                              SPA02 74
      GO TO 802                                                         SPA02 75
C                                                                       SPA02 76
C     ADAMS INTEGRATION                                                 SPA02 77
C                                                                       SPA02 78
  600 DO 601 I=1,NEQ                                                    SPA02 79
      Y1(I)=Y(I)                                                        SPA02 80
      Y(I)=Y(I)+CON*(55.0*DY(I)-59.0*DY3(I)+37.0*DY2(I)-9.0*DY1(I))     SPA02 81
      DY1(I)=DY2(I)                                                     SPA02 82
      DY2(I)=DY3(I)                                                     SPA02 83
      DY3(I)=DY(I)                                                      SPA02 84
  601 Y3(I)=Y(I)                                                        SPA02 85
      S=S+H                                                             SPA02 86
      NDIFEQ=7                                                          SPA02 87
      RETURN                                                            SPA02 88
  700 DO 701 I=1,NEQ                                                    SPA02 89
  701 Y(I)=Y1(I)+CON*(9.0*DY(I)+19.0*DY3(I)-5.0*DY2(I)+DY1(I))          SPA02 90
      GO TO 802                                                         SPA02 91
  950 Y(NV) = Y(NV)+VI
      IF (Y(NV) .LT. FV  .AND.  VI .GE. 0)     GO TO 951
      IF (Y(NV) .GT. FV  .AND.  VI .LT. 0)     GO TO 951
      S = 99.0
  951 RETURN
      END                                                               SPA02 92

C***********************************************************************
      SUBROUTINE BDYGEN (NXBODY,RADIUS,LBODY,NSEG,XEND,COEF,T,TC,TX,    SPA03  1
     1 ALPHA,IBS)                                                            SPA0

C SUBROUTINE TO CALCULATE SUPERSONIC LINE SOURCES AND DOUBLETS TO GIVE ASPA03  4
C REQUIRED BODY SHAPE AND ANGLE OF ATTACK.                              SPA03  5
C                                                                       SPA03  6
      DIMENSION XBODY(101),RBODY(101),RPBODY(101),DRDX(101),T(100)      SPA03  7
      DIMENSION TC(100),TX(101)                                         SPA03  8
      DIMENSION A(100),XF(100),RF(100)                                  SPA03  9
      DIMENSION  XEND(IBS),COEF(IBS,7)
C                                                                       SPA03 10
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ                    SPA03 11
      COMMON /SRCE/ XFIELD,RFIELD,U,V,VT                                SPA03 12
                                                                        SPA03 13
         ..  LBODY                                                      SPA03 14
                                                                        SPA03 15
        ..MAT(/// 5X,108H***** AT BASE OF BODY RADIAL DISTANCE TO MACH COSPA03 16
     ...E EMANATING FROM BODY NOSE IS LESS THAN MAXIMUM BODY RADIUS/13X, SPA03 17
```

```
     237HCHECK BODY INPUT DATA AND MACH NUMBER)
  701 FORMAT(/35X,10HBODY SHAPE,40X,11HSINGULARITY,5X,6HSOURCE,9X,
     17HDOUBLET/87X,6HORIGIN,7X,9HCONSTANTS,6X,9HCONSTANTS/24X,5HX, FT,
     210X,5HR, FT,10X,5HDR/DX,20X,1HN,7X,8HX(N), FT,8X,4HK(N),11X,
     35HKD(N))
  702 FORMAT(15X,3(0PF15.5),15X,I5,F15.5,2(1PE15.5))
C
      IF(BETA*RADIUS.LT.LBODY) GO TO 20
      WRITE (6,700)
      STOP
   20 N=NXBODY-1
      XBODY(1)=0.0
      XBODY(NXBODY)=LBODY
C
C SETUP OF POINTS ON BODY AXIS BY DIVIDING BODY LENGTH INTO N  EQUAL
C     LENGTH SEGMENTS
C
      DEL=LBODY/N
      DO 33 I=2,N
   33 XBODY(I)=XBODY(I-1)+DEL
C
C CALCULATION OF RADII AND RPBODY TO GO WITH THE AXIS POINTS

      OLBOD=1./LBODY
         35 I=1,NXBODY
         =XBODY(I)*OLBOD
        L SHAPE(XOL,NSEG,XEND,COEF,ROL,RPBODY(I),IBS)
         DY(I)=ROL*LBODY

   CALCULATION OF DRDX AND RADIUS AT MIDPOINTS BETWEEN AXIS POINTS

      DO 36 I=1,N
      XF(I)=.5*(XBODY(I)+XBODY(I+1))
      XOL=XF(I)*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,COEF,ROL,DRDX(I),IBS)
   36 RF(I)=ROL*LBODY

  THIS LOOP DETERMINES THE LOCATIONS OF THE ORIGINS OF THE CONICAL LINE
  SOURCES AND DOUBLETS.  THE ORIGINS ARE LOCATED AT THE TX(I).

      DO 10 I=1,NXBODY
   10 TX(I)=XBODY(I)-BETA*RBODY(I)
C
C ROUTINE FOR DROPPING CONTROL POINTS IF THEY ARE OUTSIDE THE MACH
C CONE OF THE NOSE.
C
  100 IF(BETA*RF(1).LT.XF(1)) GO TO 199
       N=N-1
         DY=NXBODY-1
         101 I=2,NXBODY
          (I)=XBODY(I+1)
          (I)=RBODY(I+1)
         DY(I)=RPBODY(I+1)
  101    (I)=TX(I+1)
      DO 102 I=2,N
      XF(I)=XF(I+1)
      RF(I)=RF(I+1)
  102 DRDX(I)=DRDX(I+1)
      XF(1)=.5*(XBODY(2)+XBODY(1))
      XOL=XF(1)*OLBOD
      CALL SHAPE(XOL,NSEG,XEND,COEF,ROL,DRDX(1),IBS)
      RF(1)=ROL*LBODY
        TO 100
```

```
  199 CONTINUE
C
C     DETERMINATION OF SOURCE STRENGTHS AT CONTROL POINTS MIDWAY BETWEEN
C      BODY DEFINITION POINTS.
C
C CALCULATION OF THE FIRST SOURCE STRENGTH.
C
      XFIELD=XF(1)
      RFIELD=RF(1)
      SLOPE=DRDX(1)
      CALL SOURCE(TX(1))
      A(1)=V-SLOPE*U
      T(1)=DRDX(1)/A(1)
C
C CALCULATION OF THE REST OF SOURCE STRENGTHS
C
      DO 210 I=2,N
      XFIELD=XF(I)
      RFIELD=RF(I)
      SLOPE=DRDX(I)
      DO 205 J=1,I
      CALL SOURCE(TX(J))
  205 A(J)=V-SLOPE*U
      SUM=0.
      IM1=I-1
      DO 201 J=1,IM1
  201 SUM=T(J)*A(   J)+SUM
  210 T(I)=(DRDX(I)-SUM)/A(I)
      T(NXBODY)=0.
C
C     DETERMINATION OF DOUBLET STRENGTHS AT CONTROL POINTS MIDWAY
C      BETWEEN BODY DEFINITION POINTS
C
C CALCULATION OF THE FIRST DOUBLET STRENGTH.
C
      XFIELD=XF(1)
      RFIELD=RF(1)
      SLOPE=DRDX(1)
      CALL DOUBLT(TX(1))
      A(1)=SLOPE*U-V
      TC(1)=ALPHA/A(1)
C
C CALCULATION OF THE REST OF THE DOUBLET STRENGTHS
C
      DO 215 I=2,N
      XFIELD=XF(I)
      RFIELD=RF(I)
      SLOPE=DRDX(I)
      DO 212 J=1,I
      CALL DOUBLT(TX(J))
  212 A(   J)=SLOPE*U-V
      SUM=0.
      IM1=I-1
      DO 203 J=1,IM1
  203 SUM=TC(J)*A(   J)+SUM
  215 TC(I)=(ALPHA-SUM)/A(I)
      TC(NXBODY)=0.
C
C PRINT OUT OF BODY CHARACTERISTICS
C
      WRITE (6,701)
      DO 800 I=1,NXBODY
      II=I-1
```

```
600 WRITE(6,702)   XBODY(I),RBODY(I),RPBODY(I),II,TX(I),T(I),TC(I)      SPA03144
    RETURN                                                              SPA03145
    END                                                                 SPA03146

C*******************************************************************
      SUBROUTINE BRIT (XM,AR,TR,SWEEP,CLA,XCPCB)

C     TAKEN FROM THE BRITISH DATA SHEETS  S.01.03.06
C                                         S.01.03.05
C                                         S.01.03.04
C                                         S.01.03.03
C                                         S.08.01.02
C     XM    = FREE STREAM MACH NO.
C     AR    = ASPECT RATIO (TWO WINGS JOINED WITHOUT BODY)
C     TR    = TAPER RATIO
C     SWEEP = WING MID-CHORD SWEEPBACK ANGLE (DEG)
C     CLA   = LIFT COEFF. SLOPE, BASED ON EXPOSED WING AREA  1/RAD.
C     XCPCB = CENTER OF PRESSURE (PERCENT MEAN GEOMETERIC CHORD),
C             MEASURED FROM LEADING EDGE MEAN GEOMETERIC CHORD.
C
      DIMENSION XT1(15),YT1(7),ZT1(4),XT2(8),YT2(5),ZT2(4),YT4(7)
      DIMENSION BAR1(15,7,4), BAR2(8,5,4),BAR3(15,7,4),BAR4(8,7,4)
C
      DATA XT1/0.,.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,7./
      DATA YT1/0.,1.,2.,3.,4.,5.,6./
      DATA ZT1/0.,.25,.5,1./
      DATA XT2/0.,1.,2.,3.,4.,5.,6.,7./
      DATA YT2/0.,1.,2.,3.,4./
      DATA YT4/0.,1.,2.,3.,4.,5.,6./
      DATA ZT2/0.,.25,.5,1./
      DATA (((BAR1(I,J,K),I=1,15),J=1,7),K=1,2) /
     A                 1.56,1.52,1.41,1.30,1.20,1.11,1.03,.95,.89,.83,
     1.78,.74,.70,.66,.63,1.56,1.52,1.40,1.27,1.17,1.08,1.00,.93,.87,
     2.82,.77,.73,.69,.65,.62,1.56,1.47,1.30,1.17,1.08,1.01,.95,.89,.84,
     3.79,.75,.71,.67,.64,.61,1.26,1.21,1.13,1.05,.99,.93,.88,.83,.79,
     4.75,.71,.68,.65,.62,.59,1.05,1.01,.98,.94,.89,.85,.81,.77,.73,.7,
     5.67,.64,.61,.59,.57,.90,.88,.86,.83,.80,.77,.74,.71,.68,.66,.63,
     6.61,.59,.57,.55,.79,.78,.77,.75,.73,.70,.68,.65,.63,.61,.59,.57,
     7.55,.53,.52,1.57,1.53,1.45,1.34,1.25,1.16,1.07,1.,.93,.875,.82,.78
     8,.735,.695,.66,1.57,1.53,1.42,1.31,1.20,1.11,1.03,.97,.91,.86,.81,
     9.765,.72,.685,.65,1.57,1.45,1.31,1.205,1.12,1.05,.98,.925,.87,.82,
     1.78,.74,.70,.665,.63,1.26,1.19,1.12,1.06,1.00,.95,.90,.85,.81,.77,
     2.74,.70,.67,.64,.61,1.05,1.00,.96,.92,.885,.86,.83,.79,.76,.725,
     3.69,.665,.64,.61,.59,.90,.855,.82,.80,.78,.76,.74,.72,.70,.67,.64,
     4.62,.60,.57,.55,.79,.75,.73,.71,.70,.68,.67,.65,.64,.62,.60,.58,
     5.56,.55,.53/
      DATA (((BAR1(I,J,K),I=1,15),J=1,7),K=3,4) /
     A                 1.58,1.53,1.45,1.35,1.25,1.16,1.07,1.00,.93,.88,
     1.82,.78,.73,.69,.66,1.58,1.52,1.42,1.31,1.21,1.12,1.04,.98,.915,
     2.86,.805,.76,.72,.68,.64,1.58,1.44,1.32,1.20,1.12,1.05,.98,.92,
     3.87,.82,.78,.74,.70,.76,.63,1.26,1.20,1.13,1.06,1.01,.96,.90,.85,
     4,.77,.74,.70,.66,.63,.60,1.05,1.01,
                           .97,.93,.89,.85,.82,.78,.75,.72,.68,.66,.63,
       .58,.90,.87,.84,.81,.79,.76,.73,.71,.68,.66,.64,.62,.59,.57,
      ,.79,.76,.74,.71,.69,.67,.66,.64,.62,.60,.59,.57,.55,.54,.52,
      ,1.52,1.45,1.34,1.22,1.12,1.04,.97,.90,.84,.77,.74,.70,.67,
     ,1.57,1.51,1.42,1.30,1.18,1.08,1.00,.94,.88,.82,.78,.73,.69,
     1.65,.63,1.57,1.48,1.36,1.22,1.10,1.02,.95,.88,.82,.78,.73,.69,.66,
     2.63,.61,1.27,1.17,1.10,1.12,.96,.91,.86,.81,.77,.73,.69,.66,.63,
     3.60,.58,1.06,.99,.94,.89,.84,.81,.78,.74,.71,.68,.65,.62,.60,.58,
     4.56,.91,.85,.80,.77,.[illegible],.72,.70,.68,.65,.62,.60,.58,.56,.55,.54,
     5.79,.74,.70,.67,.65,.63,.62,.61,.59,.58,.56,.54,.53,.52,.50/
      DATA BAR2 /
```

```
A               .250,.270,.273,.275,.274,.273,.270,.268,.380,
1.343,.327,.314,.303,.295,.290,.285,.500,.415,.378,.352,.335,.325,
2.318,.313,.505,.445,.405,.380,.365,.353,.345,.340,.505,.460,.428,
3.410,.395,.383,.375,.370,.145,.195,.225,.240,.247,.250,.250,.250,
4.262,.265,.267,.267,.267,.267,.267,.267,.380,.332,.305,.290,.283,
5.282,.282,.282,.395,.355,.330,.315,.305,.298,.295,.293,.405,.385,
6.365,.345,.325,.315,.307,.303,.070,.185,.225,.237,.243,.245,.247,
7.248,.177,.217,.235,.243,.245,.245,.245,.244,.285,.247,.243,.247,
8.250,.250,.248,.247,.305,.267,.257,.255,.255,.255,.255,.255,.317,
9.287,.275,.270,.270,.270,.270,.270,0.,.175,.225,.237,.243,.245,
1.245,.245,.085,.120,.165,.205,.220,.225,.227,.230,.165,.167,.170,
1.192,.197,.205,.210,.213,.193,.190,.189,.190,.193,.197,
        .200,.203,.200,.197,.189,.190,.193,.195,.196,.198/
 A (((BAR3(I,J,K),I=1,15),J=1,7),K=1,2) /
               1.56,1.70,1.76,1.75,1.70,1.43,1.22,1.06,.94,.85,
1.77,.71,.65,.60,.56,1.56,1.735,1.76,1.62,1.48,1.35,1.24,1.09,.96,
2.86,.78,.715,.66,.61,.565,1.56,1.535,1.47,1.395,1.31,1.22,1.14,
31.07,1.005,.90,.805,.73,.67,.62,.58,1.26,1.30,1.275,1.233,1.17,
41.11,1.05,1.00,.945,.90,.86,.775,.70,.645,.60,1.046,1.07,1.085,
51.085,1.06,1.023,.986,.95,.90,.86,.822,.79,.76,.682,.635,.985,
6.915,.93,.94,.94,.93,.91,.888,.855,.822,.788,.76,.73,.707,.682,
7.79,.795,.80,.80,.80,.802,.805,.81,.81,.78,.755,.73,.71,.68,.661,
81.57,1.56,1.85,1.82,1.64,1.41,1.21,1.06,.94,.845,.77,.71,.65,.60,
9.558,1.57,1.50,1.83,1.72,1.585,1.41,1.215,1.065,.955,.86,.78,.716,
1.66,.61,.562,1.57,1.13,1.46,1.46,1.40,1.305,1.22,1.11,.985,.89,
2.804,.73,.67,.62,.572,1.26,1.18,1.10,1.23,1.275,1.20,1.125,1.065,
1.015,.94,.85,.772,.71,.65,.60,1.05,1.05,1.05,1.05,1.05,1.04,
41.01,.962,.92,.87,.815,.74,.68,.63,.90,.908,.915,.923,.93,.942,
5.95,.935,.91,.875,.84,.815,.78,.73,.67,.79,.786,.78,.78,.779,.78,
6.785,.80,.82,.83,.82,.785,.755,.73,.704/
 DATA (((BAR3(I,J,K),I=1,15),J=1,7),K=3,4) /
A
               1.58,1.88,1.92,1.80,1.59,1.36,1.17,1.03,.92,.84,
1.75,.69,.635,.59,.55,1.58,1.80,1.90,1.76,1.55,1.355,1.18,1.045,
2.935,.847,.77,.705,.645,.60,.56,1.58,1.46,1.35,1.445,1.43,1.32,
31.20,1.08,.97,.87,.79,.72,.66,.61,.57,1.26,1.25,1.235,1.22,1.26,
1.21,1.145,1.08,.995,.91,.83,.756,.693,.64,.59,1.05,1.05,1.055,
 55,1.055,1.07,1.05,1.015,.97,.925,.86,.80,.74,.68,.63,.90,.90,
 .903,.905,.907,.91,.91,.90,.88,.86,.83,.77,.71,.67,.79,.792,
 .80,.804,.81,.813,.816,.82,.822,.817,.798,.76,.74,.69,1.57,
 ,2.0,1.8,1.52,1.29,1.12,.985,.88,.795,.72,.665,.62,.575,.535,
 ,1.60,2.0,1.72,1.48,1.29,1.13,1.0,.90,.81,.74,.68,.63,.583,
 5,1.57,1.55,1.52,1.48,1.44,1.29,1.16,1.03,.93,.835,.76,.70,
2.645,.60,.56,1.27,1.30,1.30,1.28,1.24,1.19,1.145,1.05,.96,.87,.80,
3.735,.68,.625,.58,1.06,1.055,1.05,1.04,1.02,.995,.98,.97,.97,.90,
4.83,.765,.70,.655,.61,.905,.90,.885,.875,.865,.86,.858,.855,.853,
5.85,.84,.785,.735,.69,.65,.795,.783,.78,.77,.762,.76,.757,.75,.75,
6 .75,.758,.76,.77,.74,.72/
 DATA BAR4 /
               .250,.350,.450,.465,.475,.483,.486,.490,.380,
1.472,.472,.472,.480,.487,.490,.495,.5,.5,.5,.5,.5,.501,.501,
2.505,.54,.54,.54,.54,.54,.525,.520,.505,.542,.580,.580,.580,.580,
 0,.558,.510,.548,.585,.625,.625,.625,.625,.625,.515,.555,.595,
 ,.673,.673,.673,.673,.145,.295,.450,.475,.485,.490,.495,.495,
 7,.350,.437,.473,.483,.487,.492,.493,.380,.445,.445,.445,.468,
 .502,.502,.502,.395,.440,.481,.481,.481,.532,.530,.523,.405,.433,
7.470,.525,.525,.525,.570,.560,.413,.457,.503,.550,.570,.570,.570,
8.615,.425,.450,.477,.510,.560,.625,.625,.625,.070,.210,.450,.477,
9.485,.490,.493,.493,.177,.270,.365,.457,.475,.482,.485,.485,.285,
1.360,.385,.415,.480,.485,.485,.485,.305,.355,.400,.435,.465,.511,
2.510,.505,.317,.323,.326,.460,.487,.501,.593,.537,.330,.320,.310,
3.425,.505,.523,.550,.585,.337,.315,.300,.423,.500,.560,.580,.605,
40.00,.330,.445,.465,.475,.483,.485,.485,.085,.310,.400,.440,.455,
5.465,.470,.470,.165,.290,.380,.425,.448,.458,.467,.475,.193,.300,
6.390,.433,.457,.470,.477,.477,.200,.330,.423,.465,.485,.493,.496,
```

```
     7.495,.200,.330,.423,.465,.485,.493,.496,.495,.200,.330,.423,.465,
     8.485,.493,.496,.495/
      X=AR*SQRT(ABS(1.-XM**2))
      Z=AR*TAN(SWEEP/57.29578)

      IF(X.GT.7.) WRITE (7,900) X
      IF(Z.GT.1.) WRITE (7,902) Z
      IF(XM.GT.1.0) GO TO 1
      CALL TABLE3 (2,CLAR,BAR1,X,XT1,15,3,Y,YT1,7,3,Z,ZT1,4,3)
      IF(Y.GT.4.) WRITE (7,901) Y
      CALL TABLE3 (2,XCPCB,BAR2,X,XT2,8,3,Y,YT2,5,3,Z,ZT2,4,3)
      CLA=CLAR*AR
      RETURN
      CALL TABLE3 (2,CLAR,BAR3,X,XT1,15,3,Y,YT1,7,3,Z,ZT1,4,3)
      IF(Y.GT.6.) WRITE (7,901) Y
      CALL TABLE3 (2,XCPCB,BAR4,X,XT2,8,3,Y,YT4,7,3,Z,ZT2,4,3)
      CLA=CLAR*AR
      RETURN
  900 FORMAT (1X,'EXTRAPOLATION REQUIRED, '10HAR'BETA = F10.4)
  901 FORMAT (1X,'EXTRAPOLATION REQUIRED, '11HAR'TANSW = F10.4)
  902 FORMAT (1X,'EXTRAPOLATION REQUIRED, TAPER RATIO = 'F10.4)
      END
C*********************************************************************
      SUBROUTINE CEL1 (RES,AK,IER)
C
C     ...............................................................
C
C        SUBROUTINE CEL1
C
C        PURPOSE
C           CALCULATE COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND
C
C        USAGE
C           CALL CEL1(RES,AK,IER)
C
C        DESCRIPTION OF PARAMETERS
C           RES    - RESULT VALUE
C           AK     - MODULUS (INPUT)
C           IER    - RESULTANT ERROR CODE WHERE
C                    IER=0  NO ERROR
C                    IER=1  AK NOT IN RANGE -1 TO +1
C
C        REMARKS
C           FOR AK=+1,-1 THE RESULT IS SET TO 1.E30.
C           FOR MODULUS AK AND COMPLEMENTARY MODULUS CK,
C           EQUATION AK*AK+CK*CK=1.0 IS USED.
C           AK MUST BE IN THE RANGE -1 TO +1
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           CEL1(AK)=INTEGRAL(1/SQRT((1+T*T)*(1+(CK*T)**2)), SUMMED
C           OVER T FROM 0 TO INFINITY).
C           EQUIVALENT ARE THE DEFINITIONS
C           CEL1(AK)=INTEGRAL(1/(COS(T)SQRT(1+(CK*TAN(T))**2)),SUMMED
C           OVER T FROM 0 TO PI/2),
C           CEL1(AK)=INTEGRAL(1/SQRT(1-(AK*SIN(T))**2),SUMMED OVER T
C           FROM 0 TO PI/2), WHERE K=SQRT(1.-CK*CK).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
```

```
            REFERENCE
            R.BULIRSCH, *NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS
            AND ELLIPTIC FUNCTIONS*, HANDBOOK SERIES SPECIAL FUNCTIONS,
C           NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C
C
      IER=0
C
C     TEST MODULUS
C
      GEO=1.-AK*AK
      IF(GEO)1,2,3
    1 IER=1
      RETURN
C
C     SET RESULT VALUE =OFLOW
C
    2 RES=1.E30
      RETURN
    3 GEO=SQRT(GEO)
      ARI=1.
    4 AARI=ARI
      TEST=AARI*1.E-4
      ARI=GEO+ARI
C
      TEST OF ACCURACY

      (AARI-GEO-TEST)6,6,5
       =SQRT(AARI*GEO)
       =0.5*ARI
        TO 4
      RES=3.14159265/ARI
      RETURN
      END

C**********************************************************************
      SUBROUTINE CEL2 (RES,AK,A,B,IER)

C     ..................................................................

         SUBROUTINE CEL2

         PURPOSE
            COMPUTES THE GENERALIZED COMPLETE ELLIPTIC INTEGRAL OF
C           SECOND KIND.
C
C        USAGE
C           CALL CEL2(RES,AK,A,B,IER)
C
         DESCRIPTION OF PARAMETERS
            RES    - RESULT VALUE
            AK     - MODULUS (INPUT)
            A      - CONSTANT TERM IN NUMERATOR
            B      - FACTOR OF QUADRATIC TERM IN NUMERATOR
            IER    - RESULTANT ERROR CODE WHERE
                     IER=0  NO ERROR
                     IER=1  AK NOT IN RANGE -1 TO +1

         REMARKS
C           FOR AK = +1,-1 THE RESULT VALUE IS SET TO 1.E30 IF B IS
C           POSITIVE, TO -1.E30 IF B IS NEGATIVE.
```

```
C              SPECIAL CASES ARE
C              K(K) OBTAINED WITH A = 1, B = 1
C              E(K) OBTAINED WITH A = 1, B = CK*CK WHERE CK IS
C              COMPLEMENTARY MODULUS.
C              B(K) OBTAINED WITH A = 1, B = 0
C              D(K) OBTAINED WITH A = 0, B = 1
C              WHERE K, E, B, D DEFINE SPECIAL CASES OF THE GENERALIZED
C              COMPLETE ELLIPTIC INTEGRAL OF SECOND KIND IN THE USUAL
C              NOTATION, AND THE ARGUMENT K OF THESE FUNCTIONS MEANS
C              THE MODULUS.
C
C           SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C              NONE
C
C           METHOD
C              DEFINITION
C              RES=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T))
C              SUMMED OVER T FROM 0 TO INFINITY).
C              EVALUATION
C              LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C              REFERENCE
C              R.BULIRSCH, *NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS
C              AND ELLIPTIC FUNCTIONS*, HANDBOOK SERIES SPECIAL FUNCTIONS,
C              NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ..................................................................
C
C
      IER=0
C
C        TEST MODULUS
C
      GEO=1.-AK*AK
      IF(GEO)1,2,6
    1 IER=1
      RETURN
C
C        SET RESULT VALUE = OVERFLOW
C
    2 IF(B)3,5,4
    3 RES=-1.E30
      RETURN
    4 RES=1.E30
      RETURN
    5 RES=A
      RETURN
C
C        COMPUTE INTEGRAL
C
    6 GEO=SQRT(GEO)
      ARI=1.
      AA=A
      AN=A+B
      W=B
    7 W=W+AA*GEO
      W=W+W
      AA=AN
      AARI=ARI
      ARI=GEO+ARI
      AN=W/ARI+AN
C
C        TEST OF ACCURACY
C
```

```
      (AARI-GEO-1.E-4*AARI)9,9,8
      =SQRT(GEO*AARI)
      )=GEO+GEO
      GO TO 7
    9 RES=.78539816*AN/ARI
      RETURN
      END

C*******************************************************************
      SUBROUTINE CROSS (FMACH,ALPHA,XLOD,CDCP,ETA)

C       CIRC. CYL. CROSS FLOW DRAG COEF. AND ETA (LENGTH) CORRECTION
C         ** ALPHA MUST BE IN RADIANS **

      DIMENSION TMCR(16),TCDCP(16),TLODE(11),TETA(11)

      DATA (TMCR(I), I=1,16)/ 0.0,.2,.3,.4,.5,.6,.7,.8,.9,1.0,1.1,
     1  1.3,1.5,1.7,1.9,2.0/
      DATA (TCDCP(I), I=1,16)/ 1.18,1.18,1.2,1.25,1.35,1.53,1.73,
     1  1.81,1.82,1.79,1.74,1.6,1.47,1.37,1.31,1.28/
      DATA (TLODE(I), I=1,11)/ 2.0,4.0,6.0,8.0,10.0,14.0,18.0,24.0,
     1  30.0,35.0,40.0/
      DATA (TETA(I), I=1,11)/ .565,.61,.639,.661,.681,.716,.745,
     1  .777,.795,.809,.820/

      XMCR= FMACH*SIN(ABS(ALPHA))
      CALL INTERP (2,1,CDCP,TCDCP,XMCR,TMCR,16,3,MIN,MAX)
      XLODE=XLOD*1.18/CDCP
      CALL INTERP (2,1,ETA,TETA,XLODE,TLODE,11,3,MIN,MAX)
      RETURN
      END

C*******************************************************************
      SUBROUTINE DBASEOPT (NDBOPT,ALFAC,MODE,ALPS,NPTS,FMACH,PHI,PSI,
     *                      XS,YS,ZS,BYY)

C          SUBROUTINE OUTPUTS DATA IN THE FORMAT REQUIRED
C          FOR USE WITH SUBMIS DATABASE PROGRAMS

      INTEGER*4 IC,IP
      CHARACTER*16  DBFN,ACON

      DIMENSION  BH(13),BYY(50,7)

      IF (NDBOPT.NE.0) GO TO 20

    5 WRITE (4,401)
  401 FORMAT(1X,'ENTER NAME OF DATA BASE OUTPUT FILE  ( [ ]SUBMIS.DAT
     1 E. MSUBMIS.DAT )')
      D (4,402) DBFN
      MAT(A16)
      EN(7,FILE=DBFN,ERR=10,STATUS='NEW',FORM='BINARY',RECL=56,
     1     BLOCKSIZE=56,ACCESS='DIRECT')
      GO TO 20

   10 WRITE (4,403) DBFN
  403 FORMAT(/1X,'FILE NAME "',A12,'" IS IN USE,...TRY AGAIN')
      GO TO 5

   20 WRITE (4,404)
  404 FORMAT(/1X,'ENTER TEST RUN NUMBER  (4 DIGITS , IE. 1001)')
      READ (4,*) NRUN
      WRITE (4,405)
```

```
  405 FORMAT(1X,'ENTER TEST CONFIGURATION  (MAX. 12 CHAR.)')
      READ (4,402) ACON

      RAD = SQRT(YS**2+ZS**2)

C***** QUADRANT I CALCULATION FOR PHIA ********************************

      PHIA = ATAN(YS/ZS)
      IF (YS.GE.0.AND.ZS.LE.0) GO TO 60

      IF (YS.LE.0.AND.ZS.LE.0) GO TO 30
      IF (YS.LE.0.AND.ZS.GE.0) GO TO 40
      IF (YS.GE.0.AND.ZS.GE.0) GO TO 50

 ...   ADRANT II CALCULATION FOR PHIA ********************************

        IA = 360-PHIA
        TO 60

 ***** QUADRANT III CALCULATION FOR PHIA *****************************

  40  PHIA = 180+PHIA
      GO TO 60

C***** QUADRANT IV CALCULATION FOR PHIA ******************************

  50  PHIA = 180-PHIA

        BH(1) = NPTS
        BH(2) = NRUN
        BH(3) = MODE
        BH(4) = FMACH

      IF (MODE.EQ.1) THEN
                IP = Y'F00FB100'
                IC = Y'800FC000'
             BH(5) = XS
             BH(6) = YS
             BH(7) = ZS
             BH(8) = RAD
             BH(9) = PHIA
         TO 70
        IF

      IF (MODE.EQ.4) THEN
                IP = Y'F007F100'
                IC = Y'020FC000'
             BH(5) = YS
             BH(6) = ZS
      ENDIF

      IF (MODE.EQ.5) THEN
                IP = Y'F00DF100'
                IC = Y'008FC000'
             BH(5) = XS
             BH(6) = YS
      ENDIF

             BH(7) = RAD
             BH(8) = PHIA
             BH(9) = ALPS

  70    BH(10) = PHI
```

```
        BH(11) = PSI
        BH(12) = ALFAC

      WRITE (7)  7,12,IC,0,IP,ACON
      WRITE (7) (BH(I), I=1,12)
        ITE (7)
         E (7)
         O I=1,NPTS
         E (7) (BYY(I,J), J=1,7)

      RETURN
      END

C***********************************************************************
      SUBROUTINE DIRCOS (A,D)

C     SUBROUTINE TO CALCULATE DIRECTION COSINES
C
      DIMENSION A(12),D(3,3)
      SPSI=SIN(A(10))
      CPSI=COS(A(10))
      STHE=SIN(A(11))
      CTHE=COS(A(11))
      SPHI=SIN(A(12))
      CPHI=COS(A(12))
      D(1,1)=CTHE*CPSI
      D(1,2)=SPHI*STHE*CPSI-CPHI*SPSI
      D(1,3)=CPHI*STHE*CPSI+SPHI*SPSI
      D(2,1)=CTHE*SPSI
      D(2,2)=SPHI*STHE*SPSI+CPHI*CPSI
      D(2,3)=CPHI*STHE*SPSI-SPHI*CPSI
      D(3,1)=-STHE
      D(3,2)=SPHI*CTHE
      D(3,3)=CPHI*CTHE
      RETURN
      END

C***********************************************************************
      SUBROUTINE DOUBLT (TX)

C SUBROUTINE TO CALCULATE THE STRENGTH OF A LINEAR LINE DOUBLET OF
C UNIT SLOPE WITH ORIGIN AT TX(J).
C
      COMMON /SRCE/ XFIELD,RFIELD,U,V,VT
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ
C
      X1=XFIELD-TX
      BR=BETA*RFIELD
      IF(X1.LE.BR    )GO TO 10
      XBR=X1/BR
      XX=SQRT(XBR*XBR-1.)
      U=BETA*XX
      ACOSH=ALOG(XBR+XX)
      XX=XBR*XX
      V=-.5*BETASQ*(ACOSH+XX)
      RETURN
C
C FIELD POINT IS AHEAD OF MACH CONE FROM DOUBLET ORIGIN.
C
   10 U=0.
      V=0.
      RETURN
      END
```

```
C*******************************************************************
      SUBROUTINE ELI1 (RES,X,CK)

      ...............................................................
C
C        SUBROUTINE ELI1
C
C        PURPOSE
C           COMPUTES THE ELLIPTIC INTEGRAL OF FIRST KIND
C
C        USAGE
C           CALL ELI1(RES,X,CK)
C
C        DESCRIPTION OF PARAMETERS
C           RES    - RESULT VALUE
C           X      - UPPER INTEGRATION BOUND (ARGUMENT OF ELLIPTIC
C                    INTEGRAL OF FIRST KIND)
C           CK     - COMPLEMENTARY MODULUS
C
C        REMARKS
C           MODULUS K = SQRT(1.-CK*CK).
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           RES=INTEGRAL(1/SQRT((1+T*T)*(1+(CK*T)**2)), SUMMED
C           OVER T FROM 0 TO X).
C           EQUIVALENT ARE THE DEFINITIONS
C           RES=INTEGRAL(1/(COS(T)*SQRT(1+(CK*TAN(T))**2)), SUMMED
C           OVER T FROM 0 TO ATAN(X)),
C           RES=INTEGRAL(1/SQRT(1-(K*SIN(T))**2), SUMMED OVER
C           T FROM 0 TO ATAN(X)).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R. BULIRSCH, NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND
C                  ELLIPTIC FUNCTIONS.
C                  HANDBOOK SERIES OF SPECIAL FUNCTIONS
C                  NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
C
C     ...............................................................
C
C
      IF(X)2,1,2
    1 RES=0.
      RETURN
    . IF(CK)4,3,4
    3 RES=ALOG(ABS(X)+SQRT(1.+X*X))
      GOTO 13
    4 ANGLE=ABS(1./X)
      GEO=ABS(CK)
      ARI=1.
      PIM=0.
    5 SQGEO=ARI*GEO
      AARI=ARI
        I=GEO+ARI
        .E=-SQGEO/ANGLE+ANGL.
         =SQRT(SQGEO)
         GLE)7,6,7
        .ACE 0 BY SMALL VALUE
```

```
      LE=SQGEO*1.E-8
      T=AARI*1.E-4
      ABS(AARI-GEO)-TEST)10,10,8
      =SQGEO+SQGEO
      IM=PIM+PIM
      IF(ANGLE)9,5,5
    9 PIM=PIM+3.1415927
      GOTO 5
   10 IF(ANGLE)11,12,12
   11 PIM=PIM+3.1415927
   12 RES=(ATAN(ARI/ANGLE)+PIM)/ARI
   13 IF(X)14,15,15
   14 RES=-RES
   15 RETURN
      END
 ************************************************************************
      SUBROUTINE ELI2 (R,X,CK,A,B)

C     ..................................................................
C
C        SUBROUTINE ELI2
C
C        PURPOSE
C           COMPUTES THE GENERALIZED ELLIPTIC INTEGRAL OF SECOND KIND
C
C        USAGE
C           CALL ELI2(R,X,CK,A,B)
C
C        DESCRIPTION OF PARAMETERS
C           R      - RESULT VALUE
C           X      - UPPER INTEGRATION BOUND (ARGUMENT OF ELLIPTIC
C                    INTEGRAL OF SECOND KIND)
C           CK     - COMPLEMENTARY MODULUS
C           A      - CONSTANT TERM IN NUMERATOR
C           B      - QUADRATIC TERM IN NUMERATOR
C
C        REMARKS
C           MODULUS K = SQRT(1.-CK*CK).
C           SPECIAL CASES OF THE GENERALIZED ELLIPTIC INTEGRAL OF
C           SECOND KIND ARE
C           F(ATAN(X),K) OBTAINED WITH A=1., B=1.
C           E(ATAN(X),K) OBTAINED WITH A=1., B=CK*CK.
C           B(ATAN(X),K) OBTAINED WITH A=1., B=0.
C           D(ATAN(X),K) OBTAINED WITH A=0., B=1.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           DEFINITION
C           R=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T)),
C                  SUMMED OVER T FROM 0 TO X).
C           EQUIVALENT IS THE DEFINITION
C           R=INTEGRAL((A+(B-A)*(SIN(T))**2)/SQRT(1-(K*SIN(T))**2),
C                  SUMMED OVER T FROM 0 TO ATAN(X)).
C           EVALUATION
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.
C           REFERENCE
C           R. BULIRSCH, NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS AND
C                  ELLIPTIC FUNCTIONS
C                  HANDBOOK SERIES OF SPECIAL FUNCTIONS
C                  NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.
```

```
C
C     ...............................................................
C
C        TEST ARGUMENT
      (X)2,1,2
    .
      URN
C        TEST MODULUS
    2 C=0.
      D=0.5
      IF(CK)7,3,7
    3 R=SQRT(1.+X*X)
      R=(A-B)*ABS(X)/R+B*ALOG(ABS(X)+R)
C        TEST SIGN OF ARGUMENT
    4 R=R+C*(A-B)
      IF(X)5,6,6
    5 R=-R
    6 RETURN
C        INITIALIZATION
    7 AN=(B+A)*0.5
      AA=A
      R=B
      ANG=ABS(1./X)
      PIM=0.
      ISI=0
      ARI=1.
      GEO=ABS(CK)
C        LANDEN TRANSFORMATION
    8 R=AA*GEO+R
      SGEO=ARI*GEO
      AA=AN
      AARI=ARI
C        ARITHMETIC MEAN
      ARI=GEO+ARI
C        SUM OF SINE VALUES
      AN=(R/ARI+AA)*0.5
      AANG=ABS(ANG)
      ANG=-SGEO/ANG+ANG
      PIMA=PIM
      IF(ANG)10,9,11
    9 ANG=-1.E-8*AANG
   10 PIM=PIM+3.1415927
      ISI=ISI+1
   11 AANG=ARI*ARI+ANG*ANG
      P=D/SQRT(AANG)
      IF(ISI-4)13,12,12
   12 ISI=ISI-4
   13 IF(ISI-2)15,14,14
   14 P=-P
   15 C=C+P
      D=D*(AARI-GEO)*0.5/ARI
      IF(ABS(AARI-GEO)-1.E-4*AARI)17,17,16
   16 SGEO=SQRT(SGEO)
         GEOMETRIC MEAN
         =SGEO+SGEO
         =PIM+PIMA
        =ISI+ISI
      GOTO 8
         ACCURACY WAS SUFFICIENT
   17 R=(ATAN(ARI/ANG)+PIM)*AN/ARI
      C=C+D*ANG/AANG
      GOTO 4
      END
```

```
C*********************************************************************
      SUBROUTINE EMPINI
C     SUBROUTINE TO INITIALIZE FOR EMPENNAGE CALCULATION
C
      COMMON /CFORCE/ CLMBY,CLMCF,CLMSB,CLNBY,CLNCF,CLNSB,CNBY,CNCF,
     1CNSB,CNX(40),CYBY,CYCF,CYSB,CYX(40),DC(3,3),DELX,DX,EDRDX(81),
     2ERAD(81),ESTRMX,UT(81),VAR(12),VSTORE,VT(81),WT(81),XMOM
      COMMON /CONSTS/ PI,DTOR,RTOD
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,FINROL,MSF,IPLNR,CLALPH
      COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,
     1FCONB,FCONC,CCL3(11),CCL5(11),XTAILI,DYFIN
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL
C
C     LOCATE SPANWISE CONTROL POINTS
C
      TA=MSF-1
      TB=FINSS-RADAV
      DYFIN=TB/TA
      DO 1 J=1,MSF
      TC=J-1
    1 RFIN(J)=RADAV+TC*DYFIN
C
C     LOCATE CONTROL POINTS ON FINS IN STORE COORDINATE SYSTEM
C
      IF (XTAIL.GT.0.0) XTAIL=-XTAIL
      FROLE(1)=DTOR*(FINROL+90.0)
      FROLE(2)=DTOR*(FINROL+270.0)
      FROLE(3)=DTOR*(FINROL+180.0)
      FROLE(4)=DTOR*(FINROL)
      JMAX=4
      IF (IPLNR.EQ.1) JMAX=2
      DO 2 J=1,JMAX
      SPHI=SIN(FROLE(J))
      CPHI=COS(FROLE(J))
      DO 2 K=1,MSF
      YTAIL(K,J)=RFIN(K)*SPHI
    2 ZTAIL(K,J)=-RFIN(K)*CPHI
C
C     COMPUTE CONSTANTS
C
      FCONA=CLALPH/(3.1415927*TB*TB)
      FCONB=FCONA/(2.0*ESTRMX)
      FCONC=-XTAIL-XMOM
      SHS=FINSS**2
      A4=RADAV**4
      A2=RADAV**2
      TOVPI=2.0/3.1415927
      DO 3 J=1,MSF
      R2=RFIN(J)**2
      ARG=(SHS*R2-A4)*(SHS-R2)/(SHS*R2)
    3 CCL3(J)=4.0*SQRT(ARG)
      IF (NROLL.EQ.0) RETURN
      IF (IPLNR.EQ.0) GO TO 10
C
C     COMPUTE LOADING COEFFICIENTS FOR ROLLING MOMENT CALCULATION
C
C     PLANAR EMPENNAGE
C
      TA=1.0+TOVPI*ACOS(2.0*FINSS*RADAV/(SHS+A2))
      TB=(SHS-A2)/(SHS+A2)
      DO 4 J=1,MSF
```

```
      R=RFIN(J)
      R2=R*R
      CCL5(J)= TA*(R+A2/R)*CCL3(J)*0.25
      IF (J.EQ.1.OR.J.EQ.MSF) GO TO 4
      TC=(R2+A2)*TB/(R2-A2)
      CCL5(J)=CCL5(J)+TOVPI*((R-A2/R)**2)*ALOG(TC+SQRT(TC*TC-1.0))
    4 CONTINUE
      RETURN

         CIFORM EMPENNAGE

         SQ= (SHS+A4/SHS)*0.25
         AM= 0.5*A2/CAPRSQ
        AM=ACOS(CTWGAM)
      STWGAM=SIN(TWGAM)
      TTWGAM=STWGAM/CTWGAM
      AK1=STWGAM
      CALL CEL1 (AKK1,STWGAM,IER)
      R2=RFIN(1)**2
      R4=R2*R2
      SH4=SHS*SHS
      TWTHE=ACOS(SHS*(A4+R4)/(R2*(A4+SH4)))
      CCL5(1)=4.0*TOVPI*CAPRSQ*(-CTWGAM*ALOG(CTWGAM)+0.5*AKK1*SIN(2.0*
        THE))
       SQRT(1.0-AK1*AK1)
        K=CK*CK
      CALL CEL2 (EPIO2K,AK1,1.0,CKCK,IER)
      DO 11 J=2,MSF
      R2=RFIN(J)**2
      R4=R2*R2
      CTWTHE=SHS*(A4+R4)/(R2*(A4+SH4))
      TWTHE=ACOS(CTWTHE)
      STWTHE=SIN(TWTHE)
      TTWTHE=TAN(TWTHE)
      ARG1=STWTHE/STWGAM
      ARG2=TTWTHE/TTWGAM
      ARG3=ASIN(ARG1)
      CCL5(J)=4.0*TOVPI*CAPRSQ*(CTWTHE*0.5*ALOG((1.0+ARG1)/(1.0-ARG1))
     1-CTWGAM*0.5*ALOG((1.0+ARG2)/(1.0-ARG2)))
      TANA1=TAN(ARG3)
      CALL ELI1 (FAK,TANA1,CK)
      CALL ELI2 (EAK,TANA1,CK,1.0,CKCK)
      CCL5(J)=CCL5(J)+2.0*TOVPI*CAPRSQ*(AKK1*SIN(2.0*TWTHE)-2.0*STWGAM
     1*COS(ARG3)*(AKK1*EAK-EPIO2K*FAK))
   11 CONTINUE
      RETURN
      END
C***********************************************************************
      SUBROUTINE FINCALC2 (FMACH,BETA,ESTRMX,FINSS,RADAV,CROOT,CTIP,
     *                     SWPLE,IAFBOD,XTAIL,CLALPH)
C
      ACOSH(X) = LOG(X+SQRT(X**2-1.))
      ATANH(X) = 0.5*LOG((1.+X)/(1.-X))
C
      DTR=0.0174532925
      PI = 3.141593
      PINV = 0.3183099
      CAVG = 0.5*(CROOT+CTIP)
      EXPFSS = FINSS-RADAV
      TSWPLE = TAN(SWPLE*DTR)
      SWPMC = ATAN(TSWPLE-.5*(CROOT-CTIP)/EXPFSS)
      TAPRAT = CTIP/CROOT
```

```
C
C     COMPUTE THE ASPECT RATIO
C
       ASRAT = 2.*EXPFSS/CAVG
C
      COMPUTE AREA RATIO

       ARAT = 2.*EXPFSS*CAVG/(PI*ESTRMX**2)

      COMPUTE BASIC CN ALPHA OF FIN

       CALL BRIT  (FMACH,ASRAT,TAPRAT,SWPMC,CNALFA,XCPCB)

       BASR = BETA*ASRAT
       QNTY1 = 1.+TAPRAT
       CBAR = .6666667*CROOT*(1.+TAPRAT**2/QNTY1)
       YMAC = (1.+2.*TAPRAT)/(3.*QNTY1*EXPFSS)

       XTAIL = XTAIL+YMAC*TSWPLE+XCPCB*CBAR
       CLALPH = CNALFA*ARAT
C
       SLNRAT = BASR*QNTY1*(TSWPLE/BETA+1.)
       IF (SLNRAT.LE.2.) GO TO 100
C
C     COMPUTE SLENDER BODY LIFT ON FIN DUE TO BODY
C
       SRB=FINSS/RADAV
       RBOS = 1./SRB
       QNTY2 = 1.-RBOS
C
       PRE1=1+(1/SRB)**4
       PRE2=0.5*ATAN(0.5*(SRB-1/SRB))+PI/4
       PRE3=(1/SRB)**2*((SRB-1/SRB)+2*ATAN(1/SRB))
       PRE4=(1-1/SRB)**2
       AKFB=2/PI*(PRE1*PRE2-PRE3)/PRE4


      COMPUTE THE LIFT ON THE BODY DUE TO THE FINS

          1=(1-(1/SRB)**2)**2
         2=(1+(1/SRB)**4)*(0.5*ATAN(0.5*(SRB-1/SRB))+PI/4)
         3=(1/SRB)**2*((SRB-1/SRB)+2*ATAN(1/SRB))
       PRE4=(1-1/SRB)**2
       AKBF=(PRE1-(2/PI)*(PRE2-PRE3))/PRE4

       IF (SWPLE.GT.1E-6) GO TO 40
C
       QNTY3 = BASR*RBOS/QNTY2
       IF (IAFBOD.EQ.0) GO TO 5

         SUBEQ1 = 1./(BASR-.5)
         SUBEQ2 = (1.+QNTY3)**2*ACOS(BASR/(BASR+SRB-1.))
         SUBEQ3 = QNTY3**2*ACOSH(1.+1./QNTY3)
         SUBEQ4 = SQRT(1.+2.*QNTY3)
         AKBFLT = PINV*SUBEQ1**(SUBEQ2-SUBEQ3+SUBEQ4-2.570796)/ARAT
       GO TO 90
C
C     COMPUTE THE LINEAR THEORY VALUE OF THE LIFT ON THE FIN IN THE
C     PRESENCE OF THE BODY
C
    5  RECFIN = QNTY3

           POSSIBILITIES EXIST: (1) 0.LE.RECFIN.LE.0.95
```

```
C                                   (2) 0.95.LE.RECFIN.LT.0
C                                   (3) 1.0.LE.RECFIN

      (RECFIN.GE.0.AND.RECFIN.LE.0.95)GO TO 10
      RECFIN.GT.0.95.AND.RECFIN.LT.1.0)GO TO 20
      (RECFIN.GT.1.0)GO TO 30

    CASE (1)

 10   PRE1=1/(PI*(1-1/(2*BASR))*ARAT*(SRB-1))
      PRE2=2*ACOS(RECFIN)+1/RECFIN-RECFIN*LOG(1/RECFIN+SQRT
     1(1/RECFIN**2-1))
      SUMM=1.0
      SKN=1.0
C
      DO 15 N=1,100
      SKN=SKN*(2*N-1)/(2*N)
      TE=SKN*RECFIN**(2*N)
      SUMM=SUMM+TE
      IF(TE.LT.0.000001)GO TO 16
 15   CONTINUE
 16   PRE3=(1/RECFIN)*(1-RECFIN*RECFIN)*SUMM
      AKBFLT=PRE1*(PRE2-PRE3)
      GO TO 90
C
    CASE (2)

      FLT=(0.334921-0.016611*((RECFIN-0.95)/0.05))/((1.0-1.0/
      .0*BASR))*ARAT*(SRB-1.0))
      O 90

       (3)

 30   AKBFLT=0.31830989/((1-1/(2*BASR))*ARAT*(SRB-1)*RECFIN)
      GO TO 90
C
 40   BCLALF = BETA*CLALPH
      BETAM = BETA/TSWPLE
      SQRTBM = SQRT(BETAM)
      BMINV = 1./BETAM
      Y4 = BETAM+1.
      5 = 1./QNTY4
      7 = 2.*BETA*RADAV/CROOT
      8 = 1./QNTY7
      9 = QNTY4*QNTY7
      Y10 = BETAM+QNTY9
      QNTY11 = BETAM*QNTY5
      QNTY12 = QNTY10*BMINV
      QNTY13 = SQRT(QNTY12)
C
      IF (IAFBOD.EQ.1) THEN
        IF (BETAM.GT.1.) THEN

            QNTY6 = SQRT(BETAM**2-1.)

          SUBEQ1 = BETAM/(QNTY6*QNTY1*QNTY7*(SRB-1.)*BCLALF)
          SUBEQ2 = QNTY11*QNTY12**2*ACOS((QNTY9+1.)/QNTY10)
          SUBEQ3 = QNTY6*QNTY5*(SQRT(1.+2.*QNTY7)-1.)
          SUBEQ4 = QNTY6*BMINV*QNTY7**2*ACOSH(1.+QNTY8)
          SUBEQ5 = QNTY11*ACOS(BMINV)
          AKBFLT = 8.*PINV*SUBEQ1*(SUBEQ2+SUBEQ3-SUBEQ4-SUBEQ5)

        ELSE
```

```
      SUBEQ1 = QNTY11**2/(QNTY1*QNTY7*(SRB-1.)*BCLALF)
      SUBEQ2 = QNTY13**3
      SUBEQ4 = (QNTY9*BMINV)**2*ATANH(1./QNTY13)
      AKBFLT = 16.*PINV*SUBEQ1*(SUBEQ2+QNTY13-SUBEQ4-2.)

    ENDIF
  ELSE

         QNTY14 = 1.+BETAM*QNTY8

    IF (BETAM.GT.1.) THEN

         QNTY6 = SQRT(BETAM**2-1.)

      IF (QNTY7.LT.1.) THEN

        SUBEQ1 = QNTY7/(QNTY6*BCLALF*QNTY1*(SRB-1.))
        SUBEQ2 = QNTY14**2*ACOS((BETAM+QNTY8)/QNTY14)
        SUBEQ3 = BETAM**2*QNTY8**2*ACOS(BMINV)
        SUBEQ4 = BETAM*QNTY8**2*QNTY6*ASIN(QNTY7)
        SUBEQ5 = QNTY6*ACOSH(QNTY8)
        AKBFLT = 8.*PINV*SUBEQ1*(SUBEQ2-SUBEQ3+SUBEQ4-SUBEQ5)

      ELSE

        SUBEQ1 = BETAM/(QNTY6*QNTY7*BCLALF*QNTY1*(SRB-1.))
        SUBEQ2 = .5*QNTY6
        SUBEQ3 = BETAM*ACOS(BMINV)
        AKBFLT = 8.*SUBEQ1*(SUBEQ2-PINV*SUBEQ3)

      ENDIF
    ELSE

         QNTY15 = 1./SQRTBM

      IF (QNTY7.LT.1.) THEN

           QNTY16 = QNTY8-1.
           QNTY17 = QNTY8**2
           QNTY18 = SQRT(QNTY16/QNTY14)

        SUBEQ1 = SQRTBM*QNTY7/(QNTY4*BCLALF*QNTY1*(SRB-1.))
        SUBEQ2 = QNTY14*SQRT(QNTY16*QNTY14)
        SUBEQ3 = QNTY17*SQRTBM**3
        SUBEQ4 = BETAM*QNTY17*QNTY4*(ATAN(QNTY15)-ATAN(QNTY18))
        SUBEQ5 = QNTY4*QNTY15*ATANH(SQRTBM*QNTY18)
        AKBFLT = 16.*PINV*SUBEQ1*(SUBEQ2-SUBEQ3+SUBEQ4-SUBEQ5)

      ELSE

        SUBEQ1 = SQRTBM/(QNTY4*QNTY7*BCLALF*QNTY1*(SRB-1.))
        SUBEQ2 = BETAM*QNTY4*ATAN(QNTY15)
        SUBEQ3 = SQRTBM**3
        AKBFLT = 16.*PINV*SUBEQ1*(SUBEQ2-SUBEQ3)

      ENDIF
    ENDIF
    IF

      RESULTANT LIFT CURVE SLOPE FOR THE FIN IS

90  CLALPH = CLALPH*((AKFB+AKBFLT)/(AKFB+AKBF))
```

```
  100 RETURN
      END
C*********************************************************************
      SUBROUTINE FUSEIO (INP,FRMAX)                                     SPA14   1
C                                                                       SPA14   3
C     SUBROUTINE TO INPUT AND OUTPUT FUSELAGE DATA                      SPA14   4
C        LINE SOURCE AND DOUBLET DISTRIBUTIONS ARE CALCULATED           SPA14   5
C        USING SUBROUTINE BDYGEN                                        SPA14   6
C
C
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ                    SPA14   9
      COMMON /FSGEOM/ FXEND(15),FCOEF(15,7),FLTHC,NFPOLY,NFXTYP(15),
     1                FXO(15),FRO(15),FCRO(15),FXI(15),FRI(15),FXF(15),
     2                FRF(15),FRCYL(15)
      COMMON /FSOR/FXL(101),FSS(100),FDS(100),NFSOR                     SPA14  11
  701 FORMAT(10I5)                                                      SPA14  15
  706 FORMAT(8F10.0)                                                    SPA14  16
  708 FORMAT(//10X,20HDISPENSER INPUT DATA)                             SPA14  17
  709 FORMAT(/15X,18HDISPENSER LENGTH =,F9.5,5H FEET/15X,16HMAXIMUM RADISPA14. 18
     1   =,F10.5,5H FEET)                                               SPA14  19
        MAT(/15X,38HPOLYNOMIALS SPECIFYING DISPENSER SHAPE//18X,        SPA14  20
           26HX/L OF END OF EACH SECTION/21X,7HSECTION,5X,3HX/L)        SPA14  21
        MAT(23X,I2,3X,7F10.5)                                           SPA14  22
        RMAT(/18X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTIONSPA14  23
     1/21X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,SPA14  24
     22HC7)                                                             SPA14  25
  733 FORMAT  (//15X,'DISPENSER SHAPE AS CALCULATED FROM THE INPUT POLYNSPA14  31
     1OMIALS, ORIGINS OF SOURCES AND DOUBLETS REPRESENTING THE',/20X,   SPA14  32
     2 'DISPENSER, AND VALUES OF THE SOURCE AND DOUBLET CONSTANTS')     SPA14  33
C                                                                       SPA14  34
C       INPUT FUSELAGE GEOMETRY AND CALCULATE POLYNOMIALS               SPA14  35
C         DESCRIBING SHAPE                                              SPA14  36
                                                                        SPA14  37
      IF (INP.NE.1)  READ (5,706) FLTHC,FRMAX                           SPA14  38
      WRITE (6,708)                                                     SPA14  39
      WRITE (6,709) FLTHC,FRMAX                                         SPA14  40
      IF (INP.EQ.1) GO TO 1
      READ (5,701) NFPOLY                                               SPA14  41
      READ (5,706) (FXEND(J),J=1,NFPOLY)                                SPA14  42
    1 WRITE (6,729)                                                     SPA14  45
      DO 4 J=1,NFPOLY                                                   SPA14  46
      FXEND(J) = FXEND(J)/FLTHC
    4 WRITE (6,730) J,FXEND(J)                                          SPA14  47
        ITE (6,731)                                                     SPA14  48
          J=1,NFPOLY                                                    SPA14  49
        INP.NE.1)  READ (5,701) NFXTYP(J)
        GEOMETRY  (INP,J,NFXTYP(J),FLTHC,FXO(J),FRO(J),FCRO(J),
                FXI(J),FRI(J),FXF(J),FRF(J),FRCYL(J),FCOEF,15)
        TE (6,730) J,(FCOEF(J,K),K=1,7)                                 SPA14  50
                                                                        SPA14  51
C     READ FUSELAGE PANEL DATA                                          SPA14  52
C                                                                       SPA14  53
      IF (INP.NE.1)  READ (5,701) NFSOR                                 SPA14  54
C                                                                       SPA14  61
C     CALCULATE LINE SOURCE AND DOUBLET DISTRIBUTIONS                   SPA14  62
C                                                                       SPA14  63
      NXBODY=NFSOR+1                                                    SPA14  64
      WRITE (6,733)                                                     SPA14  65
        LL BDYGEN (NXBODY,FRMAX,FLTHC,NFPOLY,FXEND,FCOEF,FSS,FDS,FXL,   SPA14  66
               ALFACR,15)                                                 SPA14
```

```
      RETURN
      END
C*********************************************************************
      SUBROUTINE GEOMETRY (INP,J,NTYPE,AL,XO,RO,CRO,XI,RI,XF,RF,RCYL,
     *                     COEF,IGS)
C
C     THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR THE NIELSEN
C     GENERAL POLYNOMIAL EQUATION FROM THE USUALLY KNOWN GEOMETRICAL
C     DATA FOR EACH TYPE OF BODY SEGMENT.
C
C     J IS THE NUMBER OF THE BODY SEGMENT,1<J<7,SET IN DO LOOP
C     NTYPE DESIGNATES THE TYPE OF SHAPE FOR EACH BODYSEGMENT,IN
C       SEQUENCE, FROM THE NOSE TO BASE;
C             1=OGIVE(CIRCULAR ARC) SECTION
C             2=CONICAL NOSE OR FRUSTUM SECTION
C             3=CYLINDRICAL SECTION
C
C     AL IS THE BODY LENGTH USED IN DEFINING THE COEFFICIENTS
C
      DIMENSION COEF(IGS,7)
      GO TO (10,20,30) NTYPE
C
   10 IF (INP.NE.1)  READ (5,*) XO,RO,CRO
C
C     THIS BRANCH OF THE SUBROUTINE HANDLES THE OGIVE SECTIONS
C     THE INPUT DATA FOR THIS CASE IS;
C        XO=X-LOCATION OF THE CENTER OF THE CIRCULAR ARC
C        RO=R-LOCATION OF THE CENTER OF THE CIRCULAR ARC
C        CRO=RADIUS OF THE CIRCULAR ARC
C
      COEF (J,1) = RO/AL
      COEF (J,2) = -1.0
      COEF (J,3) = 2.0*XO/AL
      COEF (J,4) =(CRO/AL)**2 - (XO/AL)**2
      COEF (J,5) = 0.0
      COEF (J,6) = 0.0
      IF (RO.EQ.0) THEN
         COEF (J,7)=1.0
      ELSE
         COEF (J,7)=-RO/ABS(RO)
      ENDIF
      RETURN
C
   20 IF (INP.NE.1)  READ (5,*) XI,RI,XF,RF
C
C     THIS BRANCH OF THE SUBROUTINE HANDLES THE CONICAL SECTIONS
C     THE INPUT DATA FOR THIS CASE IS;
C        XI=X-LOCATION OF THE UPSTREAM END OF THE SEGMENT
C        RI=BODY RADIUS AT THE UPSTREAM END OF THE SEGMENT
C        XF=X-LOCATION OF THE DOWNSTREAM END OF THE SEGMENT
C        RF=BODY RADIUS AT THE DOWNSTREAM END OF THE SEGMENT
C     NOTE: IF THE SEGMENT IS A CONICAL NOSE, THE VALUES OF BOTH
C           XI AND RI ARE 0.0, BUT MUST BE ENTERED
C
      COEF (J,1) = (RI*XF - RF*XI)/(AL*(XF-XI))
      DO 21 K=2,4
   21 COEF (J,K) =0.0
      COEF (J,5) = (RF-RI)/(XF-XI)
      COEF (J,6) =0.0
      COEF (J,7) =0.0
      RETURN
```

```
3    IF (INP.NE.1)  READ (5,*) RCYL

     THIS BRANCH OF THE SUBROUTINE HANDLES CYLINDRICAL SECTIONS

C    THE INPUT DATA FOR THIS CASE IS:
C      RCYL = THE RADIUS OF THE CYLINDER
C
     COEF (J,1) = RCYL/AL
     DO 31 K=2,7
 31  COEF (J,K) =0.0
     RETURN
     END

C*********************************************************************
     SUBROUTINE INTERP (IXTRP,LMT,Y,YT,X,XT,NX,NPX,MINX,MAXX)

     DIMENSION XT(NX),YT(NX)
     IF (IXTRP.EQ.1) GO TO 110
     Y=YT(1)
     IF (X.LE.XT(1)) RETURN
     Y=YT(NX)
     IF (X.GE.XT(NX)) RETURN
110  IF (LMT.EQ.2) GO TO 120
     CALL LIMIT (X,XT,NX,NPX,MINX,MAXX)
120  Y=YT(MINX)
     IF (MINX.EQ.MAXX) RETURN
     Y=0.
     DO 140 J=MINX,MAXX
     P=1.
     DO 130 I=MINX,MAXX
     IF (I.EQ.J) GO TO 130
     P=P*(X-XT(I))/(XT(J)-XT(I))
130  CONTINUE
140  Y=Y+YT(J)*P
     RETURN
     END

C*********************************************************************
     SUBROUTINE INTOST (XI,ETA,ZETA,X,Y,Z,DC)

C    SUBROUTINE TO TRANSFORM FROM INERTIAL TO STORE SYSTEM
C
     DIMENSION DC(3,3)
     X=XI*DC(1,1)+ETA*DC(2,1)+ZETA*DC(3,1)
     Y=XI*DC(1,2)+ETA*DC(2,2)+ZETA*DC(3,2)
     Z=XI*DC(1,3)+ETA*DC(2,3)+ZETA*DC(3,3)
     RETURN
     END

C*********************************************************************
     SUBROUTINE INVERS (A,NSYS,N,NMAX,MMAX)

C    SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS
C
     DIMENSION  A(NMAX,MMAX),X(11)

     SIGN=1.0
     NPI=N+1
     NMI=N-1
     NPLSY=N+NSYS
     DO 14 I=1,NMI
     IPI=I+1
     MAX=I
```

```
      AMAX=ABS(A(I,I))
      DO 10 K=IPI,N
      AKMAX=ABS(A(K,I))
      IF(AKMAX.LE.AMAX) GO TO 10
      MAX=K
      AMAX=AKMAX
   10 CONTINUE
      IF(AMAX.LT.1.0E-12) GO TO 16
      IF(MAX.EQ.I) GO TO 12
      DO 11 L=I,NPLSY
      TEMP=A(I,L)
      A(I,L)=A(MAX,L)
   11 A(MAX,L)=TEMP
      SIGN=-SIGN
   12 DO 14 J=IPI,N
      IF (A(J,I)) 30,14,30
   30 CONST=-A(J,I)/A(I,I)
      DO 13 L=I,NPLSY
   13 A(J,L)=A(J,L)+A(I,L)*CONST
   14 CONTINUE
      DO 15 I=1,N
      IF (A(I,I)) 15,16,15
   15 CONTINUE
      GO TO 18
   16 WRITE(6,100)
  100 FORMAT(///5X,24H***** MATRIX IS SINGULAR)
      STOP
   18 DO 21 I=NPI,NPLSY
      DO 20 KK=1,N
      K=NPI-KK
      X(K)=A(K,I)
      IF(K.EQ.N) GO TO 20

         +1
         =X(K)-A(K,J)*X(J)
         .NE.N) GO TO 19
         =X(K)/A(K,K)
      DO 21 J=1,N
   21 A(J,I)=X(J)
      RETURN
      END
C***********************************************************************
      SUBROUTINE LIMIT (X,XT,NX,NP,MINX,MAXX)

      THIS SUBROUTINE FINDS THE RANGE OF SUBSCRIPTS TO BE
       SIDERED FOR INTERPOLATION.

      DIMENSION XT(NX)
      NPX=NP
      IF (NPX.GT.NX) NPX=NX
      DO 25 I=1,NX
      IF (XT(I)-X) 25,55,45
   25 CONTINUE
      .................................. GREATER THAN MAX. SUBSCRIPT
   35 MAXX=NX
      MINX=NX-NPX+1
      RETURN
C     .................................. WITHIN RANGE
   45 MINX=I-NPX/2
      MAXX=MINX+NPX-1
      IF (MAXX.GT.NX) GO TO 35
      IF (MINX.GE.1) RETURN
```

```
      ................................. LESS THAN MIN. SUBSCRIPT
      MINX=1
      MAXX=NPX
      RETURN
C     ................................. NO INTERPOLATION NECESSARY
   55 MINX=I
      MAXX=I
      RETURN
      END
C*********************************************************************
      SUBROUTINE RESVEL (XB,YB,ZB,UTU,VTV,WTW)                          SPA20   1
C
C     SUBROUTINE TO CALCULATE RESULTANT VELOCITIES AT POINT (XB,YB,ZB)  SPA20   3
C     DUE TO ALL AIRCRAFT COMPONENTS EXCEPT SEPARATED STORE             SPA20   4
C     COORDINATES XB,YB,ZB AND RETURNED VELOCITIES,UTU,VTV,WTW,         SPA20   5
C     ARE IN FUSELAGE SYSTEM                                            SPA20   6
C                                                                       SPA20   7
      COMMON /FSOR/FXL(101),FSS(100),FDS(100),NFSOR                     SPA20   9
      COMMON /ICVEL/ UP,VP,WP,II,IF                                     SPA20  10
                                                                        SPA20  20
C                                                                       SPA20.21
C     TRANSLATE COORDINATES TO WING SYSTEM                              SPA20  22
C                                                                       SPA20  23
      UTU=0.0                                                           SPA20  28
      VTV=0.0                                                           SPA20  29
      WTW=0.0                                                           SPA20  30
C                                                                       SPA20  31
      CALL VELCAL(FSS,FDS,FXL,NFSOR,XB,YB,ZB,UP,VP,WP)                  SPA20  58
      UTU=UTU+UP                                                        SPA20  59
      VTV=VTV+VP                                                        SPA20  60
      WTW=WTW+WP                                                        SPA20  61
                                                                        SPA20  62
      RETURN                                                            SPA20109
      END                                                               SPA20110

C*********************************************************************
      SUBROUTINE SEMFOR                                                 SPA21   1
C
C     SUBROUTINE TO CALCULATE EMPENNAGE FORCES                          SPA21   2
C                                                                       SPA21   3
      DIMENSION UTL(11,4),VTL(11,4),WTL(11,4),VN(11,4),SARG(11)         SPA21   4
      COMMON /CFORCE/ CLMBY,CLMCF,CLMSB,CLNBY,CLNCF,CLNSB,CNBY,CNCF,    SPA21   5
     1CNSB,CNX(40),CYBY,CYCF,CYSB,CYX(40),DC(3,3),DELX,DX,EDRDX(81),    SPA21   6
     2ERAD(81),ESTRMX,UT(81),VAR(12),VSTORE,VT(81),WT(81),XMOM          SPA21   7
      COMMON /CONSTS/ PI,DTOR,RTOD                                      SPA21   8
      COMMON /EFORCE/ CNEM,CLMEM,CYEM,CLNEM,CLLEM                       SPA21   9
      COMMON /EMPCON/ RFIN(11),YTAIL(11,4),ZTAIL(11,4),FROLE(4),FCONA,  SPA21  10
     1FCONB,FCONC,CCL3(11),CCL5(11),XTAILI,DYFIN                        SPA21  11
      COMMON /EMPDAT/ FINSS,RADAV,XTAIL,FINROL,MSF,IPLNR,CLALPH         SPA21  12
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ                    SPA21  13
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                SPA21  14
      DO 1 J=1,4                                                        SPA21  16
      DO 1 K=1,MSF                                                      SPA21  17
      UTL(K,J)=0.0                                                      SPA21  18
      VTL(K,J)=0.0                                                      SPA21  19
    1 WTL(K,J)=0.0                                                      SPA21  20
      XXX=XTAIL+XMOM                                                    SPA21  21
      JMAX=4                                                            SPA21  22
      IF (IPLNR.EQ.1) JMAX=2                                            SPA21  23
                                                                        SPA21  24
         ULATE VELOCITIES IN STORE COORDINATE SYSTEM                    SPA21  25
                                                                        SPA21  26
         J=1,JMAX                                                       SPA21  28
```

```
      DO 2 K=1,MSF
      YYY=YTAIL(K,J)
      ZZZ=ZTAIL(K,J)
      CALL STTOIN (XXX,YYY,ZZZ,XI,ETA,ZETA,DC)
      XB=VAR(7)+XI
      YB=VAR(8)+ETA
      ZB=VAR(9)+ZETA
C
C     CALCULATE PERTURBATION VELOCITY FIELD
C
      CALL RESVEL(XB,YB,ZB,URES,VRES,WRES)
      CALL INTOST(URES,VRES,WRES,UTL(K,J),VTL(K,J),WTL(K,J),DC)
    2 CONTINUE
C
C     CALCULATE STORE FREE-STREAM VELOCITY COMPONENTS
C
      VXI=VINF*COS(ALFACR)
      VETA=0.0
      VZETA=VINF*SIN(ALFACR)
      IF (NGAM.EQ.1) GO TO 3
      VXI=VXI+VAR(1)
      VETA=VETA+VAR(2)
      VZETA=VZETA+VAR(3)
    3 VSTORE=SQRT(VXI**2+VETA**2+VZETA**2)
      CALL INTOST (VXI,VETA,VZETA,VX,VY,VZ,DC)

          FREE-STREAM COMPONENTS TO PERTURBATION VELOCITIES
        ULTING VELOCITIES ARE POSITIVE IN -X, +Y, AND -Z DIRECTIONS

       ATIO=VINF/VSTORE
      FA=VX/VSTORE
      FB=VY/VSTORE
      FC=VZ/VSTORE
      DO 4 J=1,JMAX
      DO 4 K=1,MSF
      UTL(K,J)=-VRATIO*UTL(K,J)+FA
      VTL(K,J)=+VRATIO*VTL(K,J)-FB
    4 WTL(K,J)=-VRATIO*WTL(K,J)+FC

      ADD IN Q AND R DAMPING

      IF (NDAMP.EQ.0) GO TO 6
      DUM=-XXX/VSTORE
      FA=VAR(6)*DUM
      FB=VAR(5)*DUM
      DO 5 J=1,JMAX
      DO 5 K=1,MSF
      VTL(K,J)=VTL(K,J)+FA
    5 WTL(K,J)=WTL(K,J)+FB

        RMINE VELOCITY COMPONENTS NORMAL TO PANELS EXCLUDING
          ROLL DAMPING CONTRIBUTION

        =-1.0
        J=1,JMAX
       I=SIN(FROLE(J))
      CPHI=COS(FROLE(J))
      DUM=-DUM
      DO 7 K=1,MSF
    7 VN(K,J)=DUM*(WTL(K,J)*SPHI-VTL(K,J)*CPHI)
C

      CALCULATE EMPENNAGE NORMAL FORCE, SIDE FORCE, PITCHING MOMENT,
           AND YAWING MOMENT
```

```
C
      DUM=-XTAIL/DX
      IT=DUM+0.5
      IT=IT+1
      VSO=VT(IT)
      WSO=WT(IT)
      ANG=DTOR*FINROL
      WO=WSO*COS(ANG)+VSO*SIN(ANG)
      VO=-WSO*SIN(ANG)+VSO*COS(ANG)
      A2=RADAV**2
      DO 8 K=1,MSF
      AS=(VN(K,1)+VN(K,2))/2.0+WO*A2/(RFIN(K)**2)
    8 SARG(K)=AS*CCL3(K)
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      EM=FCONA*SUM
         EM=-FCONB*SUM*FCONC
          =0.0
          =0.0
         (IPLNR.EQ.1) GO TO 10
           K=1,MSF
        -(VN(K,3)+VN(K,4))/2.0+VO*A2/(RFIN(K)**2)
    9 SARG(K)=BS*CCL3(K)
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CYEM=FCONA*SUM
      CLNEM=-FCONB*SUM*FCONC
   10 TA=CNEM
      TB=CYEM
      CA=COS(ANG)
      SA=SIN(ANG)
         M=TA*CA-TB*SA
         M=TA*SA+TB*CA
        -CLMEM
      T =CLNEM
      CLMEM=TA*CA-TB*SA
      CLNEM=TA*SA+TB*CA

      CALCULATE ROLLING MOMENT

      CLLEM=0.0
      IF (NROLL.EQ.0) RETURN
      DO 11 K=1,MSF
      ABU=(VN(K,1)-VN(K,2))/2.0
      IF (NDAMP.GT.0) ABU=ABU+VAR(4)*RFIN(K)/VSTORE
      IF (IPLNR.EQ.1) GO TO 12
      ABU=ABU+(VN(K,3)-VN(K,4))/2.0
      IF (NDAMP.GT.0) ABU=ABU+VAR(4)*RFIN(K)/VSTORE
   12 SARG(K)=ABU*CCL5(K)
   11 CONTINUE
      CALL SIMSON (MSF,SARG,DYFIN,SUM)
      CLLEM=-FCONB*SUM
      RETURN
      END
C*********************************************************************
      SUBROUTINE SFORCE (FMACH,ALOD,CDC)

C     SUBROUTINE TO CALCULATE AERODYNAMIC FORCES AND MOMENTS ON BODY
C
      DIMENSION DUMMY(12)
C
      COMMON /CFORCE/ CLMBY,CLMCF,CLMSB,CLNBY,CLNCF,CLNSB,CNBY,CNCF,
     1CNSB,CNX(40),CYBY,CYCF,CYSB,CYX(40),DC(3,3),DELX,DX,EDRDX(81),
     2ERAD(81),ESTRMX,UT(81),VAR(12),VSTORE,VT(81),WT(81),XMOM
```

```
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL

      IF (NGAM) 20,20,21
   20 CALL DIRCOS(VAR,DC)
      GO TO 22
   21 DUMMY(10)=VAR(10)-ATAN(VAR(2)/(VINF*COS(ALFACR)+VAR(1)))
      DUMMY(11)=VAR(11)+ATAN(VAR(3)/(VINF*COS(ALFACR)+VAR(1)))
      DUMMY(12)=VAR(12)
      CALL DIRCOS(DUMMY,DC)
   22 CONTINUE
      YY=VAR(8)
      ZZ=VAR(9)

      UPDATE CARDS FOR A ONE PARAMETER SWEEP

      DO 1 J=1,NHSEG
      UT(J)=0.0
      VT(J)=0.0
    1 WT(J)=0.0
      XX=DX
      DO 10 N=1,NHSEG
      XX=XX-DX
C
C     LOCATE POINT IN FUSELAGE SYSTEM
C
      XXX=XX+XMOM
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XB=VAR(7)+XI
      YB=VAR(8)+ETA
      ZB=VAR(9)+ZETA
C
C     CALCULATE PERTURBATION VELOCITY FIELD AND RESOLVE
C          INTO STORE COORDINATE SYSTEM
C
      CALL RESVEL(XB,YB,ZB,URES,VRES,WRES)
      CALL INTOST(URES,VRES,WRES,UT(N),VT(N),WT(N),DC)
   10 CONTINUE
C
C     CALCULATE STORE FREE-STREAM VELOCITY AND COMPONENTS
C
      IF (NGAM) 56,56,55
   55 VXI=VINF*COS(ALFACR)
      VETA=0.0
      VZETA=VINF*SIN(ALFACR)
      GO TO 57
   56 VXI=VINF*COS(ALFACR)+VAR(1)
      VETA=VAR(2)
      VZETA=VINF*SIN(ALFACR)+VAR(3)
   57 VSTORE=SQRT(VXI**2+VETA**2+VZETA**2)
      CALL INTOST (VXI,VETA,VZETA,VX,VY,VZ,DC)
C
C     ADD FREE-STREAM COMPONENTS TO PERTURBATION VELOCITIES
C     RESULTING VELOCITIES ARE IN -X, +Y, AND -Z DIRECTIONS
C
      VRATIO=VINF/VSTORE
      TA=VX/VSTORE
      TB=VY/VSTORE
      TC=VZ/VSTORE
      DO 60 N=1,NHSEG
      UT(N)=-VRATIO*UT(N)+TA
      VT(N)=VRATIO*VT(N)-TB
   60 WT(N)=-VRATIO*WT(N)+TC
```

```
C
C     ADD IN DAMPING TERMS
C
      IF (NDAMP.EQ.0) GO TO 70
      XX=-DX
      DO 65 N=1,NHSEG
      XX=XX+DX
      DUM=(XX-XMOM)/VSTORE
      VT(N)=VT(N)+VAR(6)*DUM
   65 WT(N)=WT(N)+VAR(5)*DUM
C
C     CALCULATE FORCES AND MOMENTS
C
   70 CONA=2.0/(ESTRMX**2)
      CALL CROSS (FMACH,VAR(11),ALOD,CDCP,ETA)
      CDC= CDCP*ETA
      CONB=CONA*CDC/3.1415927
      XSTOR=-DX
      CNBY=0.0
      CYBY=0.0
      CLMBY=0.0
      CLNBY=0.0
C
C     BUOYANCY FORCES AND MOMENTS
C
      TA= 0.5*DELX/ESTRMX

      DO 71 N=2,NHSEG,2
      NN=N/2
      XSTOR=XSTOR+DELX
      DUM=XMOM-XSTOR
      CONC=CONA*ERAD(N)**2
      COND=DUM*TA
      DCN=CONC*(WT(N+1)-WT(N-1))/DELX
      DCY=CONC*(VT(N+1)-VT(N-1))/DELX
      CNBY=CNBY+DELX*DCN
      CYBY=CYBY+DELX*DCY
      CNX(NN)=DCN
      CYX(NN)=DCY
      CLMBY=CLMBY+COND*DCN
      CLNBY=CLNBY+COND*DCY
   71 CONTINUE
C
C     SLENDER BODY THEORY FORCES AND MOMENTS
C
      CNSB=0.0
      CYSB=0.0
      CLMSB=0.0
      CLNSB=0.0
      XSTOR=-DX
      DO 72 N=2,NHSEGO,2
      XSTOR=XSTOR+DELX
      NN=N/2
      CONC=2.0*CONA*ERAD(N)*EDRDX(N)
      COND=(XMOM-XSTOR)*TA
      DCN=CNX(NN)+CONC*WT(N)
      DCY=CYX(NN)+CONC*VT(N)
      CNSB=CNSB+DELX*DCN
      CYSB=CYSB+DELX*DCY
      CNX(NN)=CNX(NN)+DCN
      CYX(NN)=CYX(NN)+DCY
      CLMSB=CLMSB+COND*DCN
      CLNSB=CLNSB+COND*DCY
```

```
      TINUE

      OUS CROSSFLOW FORCES AND MOMENTS

      =0.0
      F=0.0
      CF=0.0
      CLNCF=0.0
      IF (NHSEG.EQ.NHSEGO) GO TO 80
      NI=NHSEGO+1
      DO 73 N=NI,NHSEG,2
      VC=SQRT(WT(N)**2+VT(N)**2)
      XSTOR=XSTOR+DELX
      NN=N/2
      CONC=CONB*ERAD(N)*VC
      COND=(XMOM-XSTOR)/(2.0*ESTHMX)*DELX
      DCN=CONC*WT(N)
      DCY=CONC*VT(N)
      CNCF=CNCF+DELX*DCN
      CYCF=CYCF+DELX*DCY
      CNX(NN)=CNX(NN)+DCN
      CYX(NN)=CYX(NN)+DCY
      CLMCF=CLMCF+COND*DCN
      CLNCF=CLNCF+COND*DCY
   73 CONTINUE
   80 CONTINUE
      RETURN
      END

C***********************************************************************
      SUBROUTINE SHAPE (X,NS,XE,C,R,DRDX,ISS)

      SUBROUTINE TO CALCULATE LOCAL BODY RADIUS AND SURFACE SLOPE

      DIMENSION XE(ISS),C(ISS,7)
      DO 1 K=1,NS
      XL=XE(K)
      J=K
      IF (X.LE.XL) GO TO 2
    1 CONTINUE
    2 R=C(J,1)+X*C(J,5)+X*X*C(J,6)
      ARG=X*X*C(J,2)+X*C(J,3)+C(J,4)
      DRDX=C(J,5)+2.0*X*C(J,6)
      IF (ARG.LE.0.0) RETURN
      R=R+SQRT(ARG)*C(J,7)
      DRDX=DRDX+(2.0*X*C(J,2)+C(J,3))/(2.0*SQRT(ARG))*C(J,7)
      RETURN
      END

C***********************************************************************
      SUBROUTINE SIMSON (N,F,DX,SUM)

C     SIMPSON RULE SUBROUTINE
      DIMENSION F(101)
      SUM=F(1)+F(N)
      DO 1 I=2,N,2
    1 SUM=SUM+4.0*F(I)
      M=N-2
      DO 2 I=3,M,2
    2 SUM=SUM+2.0*F(I)
      SUM=DX*SUM/3.0
      RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE SOURCE (TX)                                            SPA26   1

C SUBROUTINE TO CALCULATE THE STRENGTH OF A LINEAR LINE SOURCE OF       SPA26   3
C UNIT SLOPE WITH ORIGIN AT TX(J).                                      SPA26   4
C                                                                       SPA26   5
      COMMON /SRCE/ XFIELD,RFIELD,U,V,VT                                SPA26   6
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ                    SPA26   7
C                                                                       SPA26   8
      X1=XFIELD-TX                                                      SPA26   9
      BR=BETA*RFIELD                                                    SPA26  10
      IF(X1.LE.BR) GO TO 10                                             SPA26  11
      XL=X1/BR                                                          SPA26  12
      D23= SQRT(XL*XL-1.)                                               SPA26  13
      U=-ALOG(XL+D23)                                                   SPA26  14
      V=BETA*D23                                                        SPA26  15
      RETURN                                                            SPA26  16
C                                                                       SPA26  17
C FIELD POINT IS AHEAD OF MACH CONE FROM SOURCE ORIGIN.                 SPA26  18
C                                                                       SPA26  19
   10 U=0.                                                              SPA26  20
      V=0.                                                              SPA26  21
      RETURN                                                            SPA26  22
      END                                                               SPA26  23

C***********************************************************************
      SUBROUTINE SOUTPT (NPTS,IPLT,ILVD,CDC,THA,CNT,CYT,CLMT,CLNT,CLLT, SPA27   1
     *                   PHI,PSI)

        OUTINE TO OUTPUT FORCE AND MOMENT DATA AND TRAJECTORY DATA      SPA27   3

       RACTER*16 FPLOT
      CHARACTER*5  VNAME

      COMMON /CFORCE/ CLMBY,CLMCF,CLMSB,CLNBY,CLNCF,CLNSB,CNBY,CNCF,    SPA27   5
     1CNSB,CNX(40),CYBY,CYCF,CYSB,CYX(40),DC(3,3),DELX,DX,EDRDX(81),    SPA27   6
     2ERAD(81),ESTRMX,UT(81),VAR(12),VSTORE,VT(81),WT(81),XMOM          SPA27   7
      COMMON /CONTROL/ NFU,NV
      COMMON /COUTPT/ DVAR(12),ESTLGC,EXST(81),TIME                     SPA27   8
      COMMON /EFORCE/ CNEM,CLMEM,CYEM,CLNEM,CLLEM                       SPA27   9
      COMMON /IFORCE/ NDAMP,NEMP,NGAM,NHSEG,NHSEGO,NROLL                SPA27  10
      COMMON /OUTINI/ XNOSEI,YNOSEI,ZNOSEI,XCGI,YCGI,ZCGI,XBASEI,YBASEI,SPA27  11
     1ZBASEI                                                            SPA27  12
                                                                        SPA27  14
  701 FORMAT(///5X,100(1H.)//6X,6HTIME =,F8.5,8H SECONDS)               SPA27  15
  702 FORMAT(/5X,29HFORCE AND MOMENT COEFFICIENTS/25X,2HCN,8X,2HCY,7X,  SPA27  16
     13HCLM,7X,3HCLN,7X,3HCLL/7X,8HBUOYANCY,5X,4F10.5/7X,12HSLENDER BODYSPA27  17
     2,1X,4F10.5/7X,9HCROSSFLOW,4X,4F10.5/7X,9HEMPENNAGE,4X,5F10.5/7X,  SPA27  18
     3/7X,63(1H-)/7X,5HTOTAL,8X,5F10.5)                                 SPA27  19
  703 FORMAT(/5X,31HLOAD AND VELOCITY DISTRIBUTIONS/14X,5HX, FT,6X,3HX/LSPA27  20
     1,5X,6HDCN/DX,4X,6HDCY/DX,5X,4HU/VS,6X,4HV/VS,6X,4HW/VS)           SPA27  21
      FORMAT(10X,9F10.5)                                                SPA27  22
        AT(/5X,68HLOCATION OF STORE IN DISPENSER COORDINATE SYSTEM, DISPA27  23
        ONS IN FEET/17X,26HRELATIVE TO DISPENSER NOSE,9X,28HRELATIVE SPA27  24
        ITIAL POSITION/20X,2HXF,8X,2HYF,8X,2HZF,11X,6HDEL XF,4X,6HDELSPA27  25
        X,6HDEL ZF)                                                     SPA27  26
        T(/5X,82HTRANSLATIONAL VELOCITIES AND ACCELERATIONS OF STORE SPA27  27
        DISPENSER COORDINATE SYSTEM/10X,28HRELATIVE TO DISPENSER MOTIONSPA27  28
     2/15X,3HDXF,8X,3HDYF,8X,3HDZF,7X,4HD2XF,7X,4HD2YF,7X,4HD2ZF)       SPA27  29
  707 FORMAT(/5X,75HROTATIONAL VELOCITIES AND ACCELERATIONS OF STORE IN SPA27  30
     1STORE COORDINATE SYSTEM/16X,1HP,10X,1HQ,10X,1HR,8X,4HPDOT,7X,4HQDOSPA27  31
     2T,7X,4HRDOT)                                                      SPA27  32
```

```
708 FORMAT(/5X,92HSTORE ANGULAR ORIENTATION IN DISPENSER COORDINATE SYSPA27 33
   1STEM AND RATES OF CHANGE OF THESE ANGLES/10X,56HANGLES IN DEGREES,SPA27 34
   2 RATES OF CHANGE IN RADIANS PER SECOND/15X,3HPSI,7X,5HTHETA,7X,3HPSPA27 35
   3HI,7X,4HDPSI,6X,6HDTHETA,6X,4HDPHI)                               SPA27 36
709 FORMAT(10X,4HNOSE,1X,3F10.5,5X,3F10.5)                           SPA27 37
710 FORMAT(10X,4HXMOM,1X,3F10.5,5X,3F10.5)                           SPA27 38
711 FORMAT(10X,4HBASE,1X,3F10.5,5X,3F10.5)                           SPA27 39

601 FORMAT(/20X,29HCROSSFLOW-DRAG COEFFICIENT IS,F10.5)
602 FORMAT(9X,6F11.5)
603 FORMAT(///5X,100(1H.)//6X,C5,2H =,F9.5)

      RTOD = 57.29578

    IF (IPLT.EQ.1.AND.NPTS.EQ.1) THEN

10  WRITE (4,401)
401 FORMAT(1X,'ENTER NAME OF FILE:    - .PLT')
    READ (4,402) FPLOT
     RMAT(A16)
      (8,FILE=FPLOT,ERR=20,STATUS='NEW',RECL=80,ACCESS='DIRECT')
      (8,801)
      AT(2X,'TIME (SEC)',3X,'THETA (DEG)',3X,'DTHETA (RPS)',4X,
         'Q (FPS)',9X,'CLM',/)
      TO 30

20  WRITE (4,403) FPLOT
403 FORMAT(/1X,'FILE NAME "',A12,'" IS IN USE,...TRY AGAIN')
    GO TO 10

    ENDIF
                                                                     SPA27 41
     OUTPUT INDEPENDENT VARIABLE                                     SPA27 42
                                                                     SPA27 43
       (NV.EQ.0) THEN
      WRITE (6,701) TIME                                             SPA27 44
    ELSE
      VALUE = VAR(NV)
      IF (NV.EQ.7) VNAME = 'XSMC'
      IF (NV.EQ.8) VNAME = 'YSMC'
      IF (NV.EQ.9) VNAME = 'ZSMC'
      IF (NV.GE.10) VALUE = VAR(NV)*RTOD
      IF (NV.EQ.10) VNAME = 'PSI'
      IF (NV.EQ.11) VNAME = 'THETA'
      IF (NV.EQ.12) VNAME = 'PHI'
      WRITE (6,603) VNAME,VALUE
    ENDIF
C
C     OUTPUT FORCES AND MOMENTS                                      SPA27 46
C                                                                    SPA27 47
    CNT=CNBY+CNSB+CNCF+CNEM                                          SPA27 48
    CYT=CYBY+CYSB+CYCF+CYEM                                          SPA27 49
    CLMT=CLMBY+CLMSB+CLMCF+CLMEM                                     SPA27 50
    CLNT=CLNBY+CLNSB+CLNCF+CLNEM                                     SPA27 51
    CLLT=CLLEM                                                       SPA27 52
    WRITE (6,702) CNBY,CYBY,CLMBY,CLNBY,CNSB,CYSB,CLMSB,CLNSB,CNCF,  SPA27 53
   1CYCF,CLMCF,CLNCF,CNEM,CYEM,CLMEM,CLNEM,CLLEM,CNT,CYT,CLMT,CLNT,  SPA27 54
   2CLLT                                                             SPA27 55
    WRITE (6,601) CDC

    IF (ILVD.EQ.0) GO TO 40
C                                                                    SPA27 62
C     OUTPUT LOAD AND VELOCITY DISTRIBUTIONS                         SPA27 63
```

```
C
      WRITE (6,703)
      DO 1 J=2,NHSEG,2
      K=J/2
      XL=EXST(J)/ESTLGC
    1 WRITE (6,704) EXST(J),XL,CNX(K),CYX(K),UT(J),VT(J),WT(J)
C
C     CALCULATE AND OUTPUT STORE LOCATION (NOSE, MOMENT CENTER,
C         AND BASE) IN FUSELAGE COORDINATE SYSTEM
C
   40 XXX=XMOM
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XNOSE=VAR(7)+XI
      YNOSE=VAR(8)+ETA
      ZNOSE=VAR(9)+ZETA
      DXN=XNOSE-XNOSEI
      DYN=YNOSE-YNOSEI
      DZN=ZNOSE-ZNOSEI
      XXX=-(ESTLGC-XMOM)
      CALL STTOIN (XXX,0.0,0.0,XI,ETA,ZETA,DC)
      XBASE=VAR(7)+XI
      YBASE=VAR(8)+ETA
      ZBASE=VAR(9)+ZETA
      DXB=XBASE-XBASEI
      DYB=YBASE-YBASEI
      DZB=ZBASE-ZBASEI
      DXCG=VAR(7)-XCGI
      DYCG=VAR(8)-YCGI
      DZCG=VAR(9)-ZCGI
      WRITE (6,705)
      WRITE (6,709) XNOSE,YNOSE,ZNOSE,DXN,DYN,DZN
      WRITE (6,710) VAR(7),VAR(8),VAR(9),DXCG,DYCG,DZCG
      WRITE (6,711) XBASE,YBASE,ZBASE,DXB,DYB,DZB
      IF (NV.GE.7) GO TO 50
C
C     OUTPUT STORE VELOCITIES
C
      WRITE (6,706)
      WRITE (6,602) VAR(1),VAR(2),VAR(3),DVAR(1),DVAR(2),DVAR(3)
C
C     OUTPUT STURE ACCELERATIONS
C
      WRITE (6,707)
      WRITE (6,602) VAR(4),VAR(5),VAR(6),DVAR(4),DVAR(5),DVAR(6)
C
C     OUTPUT STORE ANGULAR URIENTATION
C
   50 PSI = VAR(10)*RTOD
      THA = VAR(11)*RTOD
      PHI = VAR(12)*RTOD
      WRITE (6,708)
      WRITE (6,602) PSI,THA,PHI,DVAR(10),DVAR(11),DVAR(12)

      IF (IPLT.EQ.1)  WRITE (8,802) TIME,THA,DVAR(11),VAR(5),CLMT
  802 FORMAT(2X,F9.5,4X,F10.5,5X,E10.4,4X,E10.4,3X,F10.5)

      RETURN
      END

C**********************************************************************
      SUBROUTINE STORIO (INP)

C     SUBROUTINE TO INPUT AND OUTPUT STORE DATA
```

```
C         LINE SOURCE AND DOUBLET DISTRIBUTIONS ARE CALCULATED            SPA28   4
C         USING SUBROUTINE BDYGEN                                         SPA28   5
C                                                                         SPA28   6
C          ******************************************                     SPA28   7
C          *    SUBROUTINE LIMITED TO ONE STORE     *                     SPA28   8
C          ******************************************                     SPA28   9
C                                                                         SPA28  10
      DIMENSION  T(100),TX(101),TC(100)
C                                                                         SPA28  14
      COMMON /CONSTS/ PI,DTOR,RTOD                                        SPA28  16
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ                      SPA28  17
      COMMON /STGEOM/ SRMAX,SLTHC,XSMC,YSMC,ZSMC,SIC,SIBCR,PHI,PSI,       SPA28  21
     1                MSOR,NSPOLY,SXEND(7),SCOEF(7,7),NSXTYP(7),          SPA28  22
     .                SXO(7),SRO(7),SCRO(7),SXI(7),SRI(7),SXF(7),
                      SRF(7),SRCYL(7)
                                                                          SPA28  28
        RMAT (I5)                                                         SPA28  29
        ORMAT(///11X,16HSTORE INPUT DATA)                                 SPA28  30
  703 FORMAT(/40X,38HMAXIMUM        MOMENT REFERENCE LOCATION,4X,9HINCIDENSPA28  31
     1CE/29X,6HLENGTH,5X,6HRADIUS,9X,21HBEHIND DISPENSER NOSE,8X,         SPA28  32
     2 5HANGLE/31X,2HFT,9X,2HFT,8X,5HX, FT,6X,5HY, FT,6X,5HZ, FT,6X,      SPA28  33
     3 3HDEG)                                                             SPA28  34
  704 FORMAT(25X,6F11.5)                                                  SPA28  35
  712 FORMAT (8F10.5)                                                     SPA28  37
  713 FORMAT (//15X,34HPOLYNOMIALS SPECIFYING STORE SHAPE//18X,           SPA28  38
     1       26HX/L OF END OF EACH SECTION/21X,7HSECTION,5X,3HX/L)        SPA28  39
  714 FORMAT (23X,I2,3X,7F10.5)                                           SPA28  40
  715 FORMAT(/18X,51HCOEFFICIENTS OF POLYNOMIALS DESCRIBING EACH SECTIONSPA28  41
     1/21X,7HSECTION,5X,2HC1,8X,2HC2,8X,2HC3,8X,2HC4,8X,2HC5,8X,2HC6,8X,SPA28  42
     22HC7)                                                               SPA28  43
  716 FORMAT(/15X,      102HSTORE SHAPE AS CALCULATED FROM THE INPUT POLYSPA28  44
     1NOMIALS, ORIGINS OF SOURCES AND DOUBLETS REPRESENTING THE/20X,      SPA28  45
     253HSTORE, AND VALUES OF THE SOURCE AND DOUBLET CONSTANTS)           SPA28  46
C                                                                         SPA28  47
C                                                                         SPA28  48
C      READ IN AND WRITE STORE GEOMETRY                                   SPA28  49
                                                                          SPA28  50
                                                                          SPA28  51
        (INP.NE.1)  READ (5,712) SLTHC,SRMAX,XSMC,YSMC,ZSMC,SIC,PHI,      SPA28  53
                                 PSI
         E (6,702)                                                        SPA28  56
         E (6,703)                                                        SPA28  57
         TE (6,704) SLTHC,SRMAX,XSMC,YSMC,ZSMC,SIC                        SPA28  59
                                                                          SPA28  63
C       CALCULATE AND OUTPUT POLYNOMIALS DESCRIBING STORE SHAPE           SPA28  64
C                                                                         SPA28  65
      IF (INP.EQ.1) GO TO 10
      READ (5,701) MSOR                                                   SPA28  68
      READ (5,701) NSPOLY                                                 SPA28  69
      READ (5,712) (SXEND(K),K=1,NSPOLY)                                  SPA28  70
   10 WRITE (6,713)                                                       SPA28  73
        51 K=1,NSPOLY                                                     SPA28  74
        ND(K) = SXEND(K)/SLTHC
        TE (6,714) K,SXEND(K)                                             SPA28  75
        TE (6,715)                                                        SPA28  76
        52 K=1,NSPOLY                                                     SPA28  77
      IF (INP.NE.1)  READ (5,701) NSXTYP(K)
      CALL GEOMETRY  (INP,K,NSXTYP(K),SLTHC,SXO(K),SRO(K),SCRO(K),
     1               SXI(K),SRI(K),SXF(K),SRF(K),SRCYL(K),SCOEF,7)
   52 WRITE (6,714) K,(SCOEF(K,L),L=1,7)                                  SPA28  78
                                                                          SPA28  82
       CALCULATE LINE SOURCE AND DOUBLET DISTRIBUTION                     SPA28  83
                                                                          SPA28  84
```

```
      SIBCR= SIC*DTOR                                                   SPA28 85
      ALFAS=ALFACR+SIBCR                                                SPA28 86
      NXBODY = MSOR+1                                                   SPA28 87
      WRITE (6,716)                                                     SPA28 88
      . BDYGEN(NXBODY,SRMAX,SLTHC,NSPOLY,SXEND,SCOEF,T,TC,TX,      SPA28 89
       .S,7)
      .URN                                                              SPA28138
      END                                                               SPA28139
C***********************************************************************
      SUBROUTINE STTOIN (X,Y,Z,XI,ETA,ZETA,DC)                          SPA29  1
C     SUBROUTINE TO TRANSFORM FROM STORE TO INERTIAL SYSTEM             SPA29  2
C                                                                       SPA29  3
      DIMENSION DC(3,3)                                                 SPA29  4
      XI=X*DC(1,1)+Y*DC(1,2)+Z*DC(1,3)                                  SPA29  5
      ..=X*DC(2,1)+Y*DC(2,2)+Z*DC(2,3)                                  SPA29  6
      .A=X*DC(3,1)+Y*DC(3,2)+Z*DC(3,3)                                  SPA29  7
      ..TURN                                                            SPA29  8
      END                                                               SPA29  9
************************************************************************
      SUBROUTINE TABLE3 (IXTRP,W,WT,X,XT,NX,NPX,Y,YT,NY,NPY,Z,ZT,NZ,NPZ)

      TABLE LOOK-UP ROUTINE FOR 3 INDEPENDENT VARIABLES.
      IXTRP = 1 EXTRAPOLATE IF NECESSARY.
      IXTRP = 2 EXTRAPOLATION NOT ALLOWED. TAKE THE LAST(OR FIRST) POINT.
C     W     = ANSWER,(DEPENDENT VARIABLE CORRESPONDING TO INPUTS  X,Y,Z)
C     WT    = TABLE OF DEPENDENT VARIBALE CORRESPONDING TO XT,YT,ZT
C             WT(I,J,K)  INCREMENT SUBSCRIPTS LEFT TO RIGHT WHEN LOADING
C     X     = THE ARGUMENT OR INDEPENDENT VARIABLE  X
C     XT    = TABLE OF INDEP.  X  VALUES (MUST BE IN INCREASING ORDER)
C     NX    = NUMBER OF POINTS IN  XT
      NPX   = NUMBER OF POINTS TO USE FOR  X  INTERPOLATION
      Y     = THE ARGUMENT OR INDEPENDENT VARIABLE  Y
      YT    = TABLE OF INDEP.  Y  VALUES (MUST BE IN INCREASING ORDER)
      NY    = NUMBER OF POINTS IN  YT
      NPY   = NUMBER OF POINTS TO USE FOR  Y  INTERPOLATION
C     Z     = THE ARGUMENT OR INDEPENDENT VARIABLE  Z
C     ZT    = TABLE OF INDEP.  Z  VALUES (MUST BE IN INCREASING ORDER)
C     NZ    = NUMBER OF POINTS IN  ZT
C     NPZ   = NUMBER OF POINTS TO USE FOR  Z  INTERPOLATION

      DIMENSION  XT(NX),YT(NY),ZT(NZ),WT(NX,NY,NZ),B(20),A(10)

      CALL LIMIT (Z,ZT,NZ,NPZ,MINZ,MAXZ)
      CALL LIMIT (Y,YT,NY,NPY,MINY,MAXY)
      CALL LIMIT (X,XT,NX,NPX,MINX,MAXX)
      DO 41 K=MINZ,MAXZ
      DO 42 J=MINY,MAXY
42    CALL INTERP (IXTRP,2,B(J),WT(1,J,K),X,XT,NX,NPX,MINX,MAXX)
41    CALL INTERP (IXTRP,2,A(K),B,Y,YT,NY,NPY,MINY,MAXY)
      CALL INTERP (IXTRP,2,W,A,Z,ZT,NZ,NPZ,MINZ,MAXZ)
      RETURN
      END
C***********************************************************************
      SUBROUTINE VELCAL (T,TC,TX,N,XP,Y,ZP,U1,V1,W1)                    SPA35  1

C SUBROUTINE TO CALCULATE THE VELOCITIES FOR THE FIELD POINT X,Y,Z DUE  SPA35  3
C TO THE BODY SINGULARITIES.                                            SPA35  4
C                                                                       SPA35  5
C     XP,Y,ZP - FIELD POINT COORDINATES WHERE VELOCITY                  SPA35  6
```

```
                IS TO BE CALCULATED
      XP,Y,ZP IN BODY SYSTEM
            XP - POSITIVE FORWARD
            Y  - POSITIVE RIGHT
            ZP - POSITIVE DOWN
      U1,V1,W1 - VELOCITIES ACCORDING TO THE FOLLOWING SIGN CONVENTION
C             U1 - POSITIVE FORWARD
C             V1 - POSITIVE TO RIGHT
C             W1 - POSITIVE DOWN
C
      DIMENSION T(100),TC(100),TX(101)
C
      COMMON /CONTROL/ NFU,NV
      COMMON /SRCE/ XFIELD,RFIELD,U,V,VT
      COMMON /FLOW/ ALFACR,GAMF,RHO,VINF,BETA,BETASQ
C
C     TRANSFORM FIELD POINT COORDINATES TO VELCAL SYSTEM
C
      X=-XP
      Z=-ZP
      XFIELD=X
      RFIELD=SQRT(Y*Y+Z*Z)
      THETA=ATAN2(Y,Z)
C
C CALCULATION OF AXIAL, RADIAL AND TANGENTIAL PERTURBATION VELOCITIES
C
      US=0.
      VS=0.
      UD=0.
      VD=0.
      VTD=0.

      IF (NFU.EQ.0) GO TO 115

      DO 110 J=1,N
      X1=XFIELD-TX(J)
      BR=BETA*RFIELD
      IF(X1.LE.BR) GO TO 115
      XL=X1/BR
      D23=SQRT(XL*XL-1.)
      ACOSH=ALOG(XL+D23)
      U=-ACOSH
      V=BETA*D23
      US=US+T(J)*U
      VS=VS+T(J)*V
      U=V
      XX=XL*D23
      V=-.5*BETASQ*(ACOSH+XX)
      VT=.5*BETASQ*(ACOSH-XX)
      UD=UD+U*TC(J)
      VD=VD+V*TC(J)
  110 VTD=VTD+VT*TC(J)
C
C TRANSFORMATION OF VELOCITIES INTO ORIGINAL COORDINATE SYSTEM...I.E....
C U1,V1,W1 RATHER THAN U,VR,VTHETA

        TINUE
        COSTH=COS(THETA)
        -US-UD*COSTH
        VR=VD*COSTH+VS
        SINTH=SIN(THETA)
        =VTD*SINTH
      V1=VR*SINTH+VTD*COSTH


      W1=VTD*SINTH-VR*COSTH
      RETURN
      END
```

APPENDIX C.7

INPUT DATA, SEQUENCE, FORMAT, AND DEFINITIONS FOR FORMATTED FILE OPERATION

Item 1 NCARDS FORMAT(I5)

NCARDS is the number of lines or cards of alphanumeric data used to describe the problem.

Item 2 HEAD FORMAT(20A4)

HEAD is NCARDS lines or cards of alphanumeric data selected by the user to describe the problem.

Item 3 NFU, NSTRS FORMAT(2I5)

NFU Dispenser missile present? 0 = No, 1 = Yes.

NSTRS Number of submissiles present, program limited to one (1).

Item 4 FMACH, RHO, VINF, GAMF, ALFAC FORMAT(5F10.0)

FMACH Free stream Mach number.

RHO Static air density at flight or test conditions, slugs per cubic foot.

VINF Free stream velocity, feet per second.

GAMF Dispenser missile flight path angle, degrees.

ALFAC Dispenser missile angle of attack, degrees.

Item 5 FLTHC, FRMAX FORMAT (2F10.0)

FLTHC Dispenser missile fuselage length, feet.

FRMAX Dispenser missile maximum radius, feet.

Item 6 NFPOLY FORMAT (I5)

NFPOLY Number of polynomials specifying the dispenser missile fuselage shape, $1 \leq$ NFPOLY ≤ 15.

Item 7 FXEND(J) [NFPOLY VALUES] FORMAT (8F10.0)

FXEND(J) x/ℓ of the end points of polynomials specifying the dispenser missile fuselage shape, NFPOLY values.

Items 8 and 9 are read in a Do Loop and there must be values for each item for each polynomial segment, i.e., NFPOLY sets of values. Item 9 variables depend upon the value of item 8.

Item 8 NFXTYP(J) FORMAT (I5)

NFXTYP An integer which identifies the type of polynomial defining the segment.

NFXTYP = 1 is a circular arc (ogive) segment.
NFXTYP = 2 is a cone or conical frustum segment.
NFXTYP = 3 is a cylindrical segment.

Item 9 FXO, FRO, FCRO [NFXTYP = 1] FORMAT (3F10.0)

- FXO Axial location of the center of the circular arc, feet.
- FRO Radial location of the center of the circular arc, feet.
- FCRO Radius of the circular arc, feet.

Item 9 FXI, FRI, FXF, FRF [NFXTYP = 2] FORMAT (4F10.0)

- FXI Axial location of the upstream end of the segment, feet.
- FRI Body radius at the upstream end of the segment, feet.
- FXF Axial location of the downstream end of the segment, feet.
- FRF Body radius at the downstream end of the segment, feet.

Item 9 FRCYL [NFXTYP = 3] FORMAT (F10.0)

- FRCYL Radius of the cylindrical segment, feet.

Item 10 NFSOR FORMAT (I5)

- NFSOR Number of line sources and line doublets to be used over the dispenser missile fuselage length.

Item 11 SLTHC, SRMAX, XSMC, YSMC, ZSMC, SIC, PHI, PSI FORMAT (8F10.0)

- SLTHC Submissile length, feet.
- SRMAX Submissile maximum radius, feet.
- XSMC Longitudinal distance from dispenser missile nose to the submissile moment reference center, feet; positive aft of dispenser missile nose.
- YSMC Lateral distance from dispenser missile center line to the submissile moment reference center, feet; positive to right.
- ZSMC Vertical distance from dispenser missile center line to the submissile moment reference center, feet; positive below.
- SIC Submissile pitch incidence angle relative to dispenser missile center line, degrees; positive submissile nose up toward dispenser.
- PHI Submissile roll angle with respect to the dispenser missile, degrees; positive clockwise about x-axis.
- PSI Submissile yaw angle with respect to the dispenser missile center line, degrees; positive nose to the right.

Item 12 MSOR FORMAT (I5)

- MSOR Number of line sources and line doublets to be used to model the submissile volume and angle of attack effects; MSOR $\leq$ 100.

Item 13 NSPOLY FORMAT (I5)

- NSPOLY Number of polynomials specifying the submissile shape, 1 $\leq$ NSPOLY $\leq$ 7.

Item 14 SXEND(K) FORMAT (7F10.0)

- SXEND(K) x/ℓ of the end points of the polynomials specifying submissile shape, NSPOLY values.

Items 15 and 16 are read in a Do Loop and there must be values for each item for each polynomial segment of the submissile, i.e., NSPOLY sets of values. Item 16 variables depend upon the value of item 15.

Item 15 NSXTYP (K) FORMAT (I5)

NSXTYP An integer which identifies the type of polynomial defining the segment.

NSXTYP = 1 is a circular arc (ogive) segment

NSXTYP = 2 is a cone or conical frustum segment

NSXTYP = 3 is a cylindrical segment.

Item 16 SXO, SRO, SCRO [NSXTYP = 1] FORMAT (3F10.0)

SXO Axial location of the center of the circular arc, feet.

SRO Radial location of the center of the circular arc, feet.

SCRO Radius of the circular arc, feet.

Item 16 SXI, SRI, SXF, SRF [NSXTYP = 2] FORMAT (4F10.0)

SXI Axial location of the upstream end of the segment, feet.

SRI Body radius at the upstream end of the segment, feet.

SXF Axial location of the downstream end of the segment, feet.

SRF Body radius at the downstream end of the segment, feet.

Item 16 SRCYL [NSXTYP = 3] FORMAT (F10.0)

SRCYL Radius of the cylindrical segment, feet.

Item 17 NSEG, NSEGXO, NGAM, NROLL, NEMP, NDAMP, NV FORMAT(7I5)

NSEG Number of equal length segments the submissile body is divided into for the force calculation; NSEG $\leq$ 40.

NSEGXO Number of body segments to the flow separation location.

NGAM Trajectory to simulate wind tunnel captive store trajectory? 0 = No, 1 = Yes (Does not affect the single variable sweep).

NROLL Rolling moment to be calculated? 0 = No, 1 = Yes.

NEMP Empennage present? 0 = No, 1 = Yes.

NDAMP Damping to be included in force calculation? 0 = No, 1 = Yes.

NV Variable number which controls the type of calculation to be performed.

NV = 0, separation trajectory.

NV = 7, submissile performs X-sweep.

NV = 9, submissile performs Z-sweep.

NV = 11, submissile performs α-sweep.

Item 18 SMASS, FIXX, FIYY, FIZZ, FIYZ, FIXZ, FIXY FORMAT(7F10.0)

SMASS Submissile mass, slugs

FIXX I_{XX} moment of inertia, slug-ft^2.

FIYY I_{YY} moment of inertia, slug-ft^2.

FIZZ I_{ZZ} moment of inertia, slug-ft^2.

FIYZ I_{YZ} product of inertia, slug-ft^2.

FIXZ I_{XZ} product of inertia, slug-ft^2.

FIXY I_{XY} product of inertia, slug-ft^2.

Item 19 XMOM, XBAR, YBAR, ZBAR FORMAT(4F10.0)

XMOM Location along submissile axis about which the pitching and yawing moments are to be taken, negative behind nose, feet; same point about which moments of inertia are taken.

XBAR — X location of submissile center of gravity measured from moment center, feet; positive forward.

YBAR — Y location of submissile center of gravity measured from submissile axis, feet; positive to the right.

ZBAR — Z location of submissile center of gravity measured from submissile axis, feet; positive below.

Item 20 CA FORMAT(F10.0)

CA — Submissile axial force coefficient relative to the dispenser missile axial force coefficient; reference area is submissile maximum cross-sectional area.

The next two items are input <u>only</u> if the submissile has an empennage (NEMP = 1).

Item 21 IPLNR, MSF, IAFBOD FORMAT(3I5)

IPLNR = 0, cruciform empennage.

IPLNR = 1, planar empennage.

MSF = number of spanwise control points on each fin, must be odd and $5 \leq MSF \leq 11$.

IAFBOD = 0, No afterbody aft of fin trailing edge.

IAFBOD = 1, Long afterbody aft of fin trailing edge.

Item 22 XTAIL, RADAV, FINSS, FINROL, CROOT, CTIP, SWPLE FORMAT(7F10.0)

XTAIL — X location where the fin leading edge intersects the body surface, measured from submissile nose, feet; negative number.

RADAV — Average submissile body radius in empennage region, feet; positive number.

FINSS — Tail fin semispan, measured from submissile longitudinal axis, feet; positive number.

FINROL — Initial fin orientation, degrees, $0 \leq FINROL \leq 90$; FINROL = 0 if fins vertical and horizontal.

CROOT — Tail fin root chord, feet; positive number.

CTIP — Tail fin tip chord, feet; positive number.

SWPLE — Tail fin leading edge sweep angle, degrees; positive for swept back leading edge.

Item 23 VXZERO, VYZERO, VZZERO, VAR(4), VAR(5), VAR(6) FORMAT(6F10.0)

VXZERO — Submissile initial longitudinal velocity, ft/s; positive forward.

VYZERO — Submissile initial lateral velocity, ft/s; positive to the right.

VZZERO — Submissile initial vertical velocity, ft/s; positive downward.

VAR(4) — Submissile initial roll rate, P, radians/s.

VAR(5)   Submissile initial pitch rate, Q, radians/s.
VAR(6)   Submissile initial yaw rate, R, radians/s.

Item 24   DTIME, TIMEI, TIMEF   FORMAT (3F10.0)
DTIME   Integration interval, seconds.
TIMEI   Initial time, seconds.
TIMEF   Final time, seconds.

If a trajectory is to be run either from T = 0 or from a restart time, NV = 0, and Item 24 is input. For a trajectory restart, the last time step in the previous calculation output file is used to obtain the initial conditions. A trajectory restart requires that Item 24B be input.

Item 24B   VAR[12 values]   FORMAT(8F10.0)
This item consists of two cards with VAR(1) through VAR(8) on the first card and VAR(9) through VAR(12) on the second. The table below gives the notation used to identify VAR(1) through VAR(12) on the trajectory program output.

Program Notation	Output Notation
VAR(1)	DXF, ft/s
VAR(2)	DYF, ft/s
VAR(3)	DZF, ft/s
VAR(4)	P, radians/s
VAR(5)	Z, radians/s
VAR(6)	R, radians/s
VAR(7)	XF of XMOM, ft
VAR(8)	YF of XMOM, ft
VAR(9)	ZF of XMOM, ft
VAR(10)	PSI, degrees
VAR(11)	THETA, degrees
VAR(12)	PHI, degrees

Multiple cases may be run by using sequential sets of data in the input file.

DTIME is the integration interval to be used in the integration subroutine. The required interval depends, at least to a certain extent, upon the scale of the problem. For aircraft/store sized vehicles, Reference 6 reports that values from 0.05 to 0.10 second yield satisfactory results. It is suggested that cases be run with different time intervals to verify the validity of the time interval.

If a single variable sweep is to be run, NV = 7, 9, or 11, and Item 25 must be input; however, Item 24 is omitted.

Item 25   FV,VI   FORMAT(2F10.0)
FV is the final value of the variable to be varied;
VI is the increment by which the variable is varied

The variable may be swept from a small value to a larger value using a positive increment or it may be vice versa.

APPENDIX D

OPERATIONAL PROGRAM FILENAMES

There are operational versions of each of the three programs available as Group or system files. Lists of each set of files is given below:

A. Subsonic Source Program
   1. NEARSOR.FTN/100
   2. NEARSOR.CSS/0
   3. NEARSOR.TSK/100
   4. NEARSOR.INP/100 [dispenser missile input data file]

To run this program with the sample input data file, type NEARSOR NEARSOR. For the general case, the input data filename is typed in following the program name, i.e., NEARSOR FILENAME(INPUT).

B. Subsonic Trajectory Program
   1. NEARSUB.FTN/100
   2. NEARSUB.CSS/0
   3. NEARSUB.TSK/100
   4. NEARSUB.NAM/100 [example Namelist input data file]
   5. NEARSUB.INP/100 [example formatted input data file, see Appendix B.8]
   6. DISPENSX.SRC/100 } Source files for input
   7. SUBMISAX.SRC/100 }

To run the program with the Namelist file, type NEARSUB; program prompts will request the Namelist filename and the source filenames. To run the program with the formatted file, type NEARSUB NEARSUB; program prompt will request the source filenames.

C. Supersonic Trajectory Program
   1. NEARSUP.FTN/100
   2. NEARSUP.CSS/0
   3. NEARSUP.TSK/100
   4. NEARSUP.NAM/100 [example Namelist input data file]

To run the program with the Namelist file, type NEARSUP, a program prompt will request the Namelist filename. To fun the program with a formatted file, type NEARSUP FILENAME(INPUT).